all rights reversed

HACKER

Απρίλης 2012 # Τιμή τεύχους 6,40€

τεύχος 007 - A View to a Hack Edition

Τα malware του κόσμου, τώρα στο PC σας [*]



Βγείτε Αμερική ό,τι ώρα θέλετε: Τώρα μπορείτε και σας συμφέρει!

Εξερευνήστε το σκοτεινό μέρος του web, που μέχρι χθες αγνοούσατε

Πώς φτιάχνονται αποτελεσματικές wordlists για πολύ συγκεκριμένους στόχους

[*] ...μην τα σβήσετε, μελετήστε τα!

tefhos 7o

Περιεχόμενα

και οδηγίες χρήσης

Strictly online | **πώς δουλεύουν τα δίκτυα: τι είναι το subnetting και γιατί πρέπει να το ξέρετε** | διαβάστε το στο http://deltahacker.gr/?p=5138

Εκδότης / Διευθυντής	Χρήστος Βαρελάς
Αρχισυντάκτης	Παναγιώτης Βαρελάς
Σύμβουλοι έκδοσης	Νίκος Μουρατίδης, Βούλα Παυλίδου
Σύνταξη	Παναγιώτης Βαρελάς, Χρήστος Βαρελάς, Ανδρέας Βενιέρης, Γιώργος Γιάννου, Μανώλης Κιαγιάς, Ιωάννης Κονιάρης, Γιώργος Μελέκος, Νίκος Μουρατίδης, Νικόλαος Σηφάκης, Μέσος Παπαδόπουλος (TM), Χρίστος Τόμπρας
Γραμματική επιμέλεια	Βούλα Παυλίδου
Καλλιτεχνική Επιμέλεια	Πέτρος Φιλιππίδης

Το περιοδικό deltaHacker είναι συνδρομητικό και εκδίδεται από την Parabing Creations δώδεκα φορές το χρόνο, κάθε μήνα. Η Parabing Creations έχει την έδρα της στη Μαιώτιδος 3, 55 133, Καλαμαριά. Η εκτύπωση και η βιβλιοδεσία γίνονται από την ThessPrint A.E. Το email επικοινωνίας του περιοδικού είναι το talk2us@deltahacker.gr και το email για τις συνδρομές το subscriptions@deltahacker.gr. Οι πληροφορίες για τις συνδρομές είναι στο http://deltahacker.gr/subscriptions και οι παραγγελίες γίνονται από το http://deltahacker.gr/order. Όλες οι απόψεις που εκφράζονται στα άρθρα δεν εκφράζουν απαραίτητα και τη γνώμη του περιοδικού.

(*) Ο Τσακ Νόρις *δεν* διαβάζει deltaHacker.



Είστε από εκείνους
που δεν βλέπουν τον υπολογιστή τους
ως άλλη μια οικιακή συσκευή;

?

Αν ναι, τότε είστε αναγνώστης
του περιοδικού deltaHacker
έστω κι αν δεν έχετε γίνει ακόμα!

Το deltaHacker έχει ως αποκλειστική αποστολή
να ξυπνήσει τον δημιουργικό χάκερ μέσα σας
και για να το πετύχει κυκλοφορεί 12 φορές το χρόνο
αλλά όχι στα περίπτερα.

Γίνετε τώρα συνδρομητής στο ένα και μοναδικό περιοδικό που ακονίζει το μυαλό
και κάνει τους υπολογιστές, τα δίκτυα και το hardware ξανά ενδιαφέροντα.

Χαρίστε στον εαυτό σας ή στους ανθρώπους σας μια συνδρομή στο περιοδικό

HACKER

Γεύσεις από την ύλη του περιοδικού στο **http://deltaHacker.gr**
Αγορές τευχών/συνδρομών στο **http://deltaHacker.gr/order**

http://twitter.com/deltaHacker
http://facebook.com/deltaHacker
http://gplus.to/deltaHacker

Το deltaHacker είναι αποκλειστικά συνδρομητικό, αποστέλλεται ταχυδρομικά και δεν κυκλοφορεί στα περίπτερα. Οι αγορές μεμονωμένων τευχών ή συνδρομών γίνονται δικτυακά, από το **http://deltaHacker.gr/order**



Ο συνήγορος του ροπάλου!

Ελεύθερα ρόπαλα - Ανοιχτά κεφάλια. Εχμ, συγνώμη, ανοιχτά μυαλά ήθελα να πω. Όποιος δεν ξέρει να χειρίζεται ρόπαλο είναι τουλάχιστον ανόητος, για να μην πω εκτός τόπου και χρόνου. Έλα στο μέλλον, άνθρωπέ μου! Πάρε κι εσύ ένα ρόπαλο και πάλεψέ το. Με τη δουλειά σου θ' ασχοληθείς αργότερα...

Μπορώ να φανταστώ πολλές εργασίες, οι οποίες θα ολοκληρώνονταν άνετα με τη βοήθεια ενός ρόπαλου. Στ' αλήθεια, πολλές. Ειδικά αν ο χειριστής έχει την υπομονή ενός μοναχού Σαολίν, το ρόπαλο μπορεί να ολοκληρώσει *κάθε* εργασία.

Δεν με κοπάνησαν με κανένα ρόπαλο, ούτε ξύπνησα ημίτρελος. Προτιμώ τα ρόπαλα από καιρό τώρα, για κάθε εργασία μου κι αυτό αποτελεί ώριμη επιλογή. Μου αρέσει που το καθένα είναι διαφορετικό από τ' άλλα. Κάτι σαν τους ανθρώπους ένα πράγμα, αλλά σε εργαλείο! Μαθαίνεις να δουλεύεις μ' ένα συγκεκριμένο ρόπαλο κι όταν υποχρεωθείς να το αλλάξεις, πρέπει να μάθεις σχεδόν τα πάντα από την αρχή. Σκέτη ομορφιά! Αυτό ακριβώς μου αρέσει να κάνω με τα εργαλεία μου: Να μην ξεμπερδεύω ποτέ με τις παραξενιές τους. Ν' ανακαλύπτω συνέχεια ατέλειες και φυσικά να εφευρίσκω λύσεις --workarounds-- για τα προβλήματα που παρουσιάζουν. Όσο για τη δουλειά... Ε, έλα μωρέ, θα γίνει κι αυτή κάποτε, δεν βαριέσαι. Σημασία έχει να δουλεύω με το αγαπημένο μου ρόπαλο. Ξέρω, ακούγονται τρελά όλα αυτά. Σας διαβεβαιώνω πως δεν είναι. Το 'χω σκεφτεί πολύ το θέμα με τα ρόπαλα κι έχω σοβαρούς λόγους για να τα προτιμώ, έναντι κάθε άλλου εργαλείου.

Τα ρόπαλα δεν κατασκευάζονται από πολυεθνικούς κολοσσούς. Δεν προέχονται από μεγιστάνες της αγοράς, οι οποίοι κατασκευάζουν ατελείωτα και πανομοιότυπα εργαλεία, χωρίς ψυχή. Τα ρόπαλα φτιάχνονται από εκατοντάδες τεχνίτες ανά τον κόσμο, με μεράκι και αγάπη για το "καλό ρόπαλο". Μάλιστα με τον όρο "καλό", ο καθένας από τους τεχνίτες έχει στο μυαλό του κάτι εντελώς διαφορετικό. Έτσι, προκύπτει μια τεράστια ποικιλία από μοναδικά ρόπαλα (Σ.τ.Ε. Από μοναδικές καλότητες, ίσως;). Όλο και κάποιο θα βρεις να σου ταιριάζει. Μπορεί να μη συνδυάζεται εύκολα με τα υπόλοιπα ρόπαλά σου, αλλά μόνο του, απομονωμένο από το περιβάλλον του, θα λειτουργεί άψογα. Όλα αυτά όμως έχουν μικρή σημασία. Επιτρέψτε μου να συνεχίσω, παρακαλώ...

Το ρόπαλο μπορείς να το χρησιμοποιήσεις *όπως* θες. Δεν σε περιορίζει κανένας! Καμία άδεια χρήσης, κανένας νταβατζής των εργαλείων δεν μπορεί να καθορίσει τι θα κάνεις και τι δεν θα κάνεις με το ρόπαλό σου. Πέρα μάλιστα από τις άδειες χρήσης, δεν υπάρχει ούτε καν κάποια κοινωνική σύμβαση γύρω από το πώς χρησιμοποιούνται τα ρόπαλα. Ό,τι και να κάνεις μ' ένα ρόπαλο, κανένας δεν θα μπορεί να σε κατηγορήσει -- ή έστω να σε κοιτάξει στραβά. Εδώ που τα λέμε, τα ρόπαλα θα μπορούσαν ν' αποτελούν το υπέρτατο σύμβολο της ανθρώπινης ελευθερίας. Έχεις ρόπαλο; Είσαι ελεύθερος. Χρησιμοποιείς κάθε φορά ένα κατάλληλα μελετημένο, και προσεκτικά κατασκευασμένο σφυρί, κλειδί ή κατσαβίδι; Είσαι θύμα των πολυεθνικών -- κορόιδο!

Μπορεί να μετατρέψεις το ρόπαλο σ' ότι θες. Κανένας δεν θα παραξενευτεί. Ξέρετε κανέναν ο οποίος να μετέτρεψε το σφυρί του σε τανάλια; Μήπως είδατε κάποιον να χρησιμοποιεί το κατσαβίδι σαν

μέγγενη; Φυσικά και όχι. Όλα αυτά τα τυποποιημένα εργαλεία κατασκευάζονται από μεταλλικά κράματα, τα οποία είναι πρακτικά αδύνατο να επεξεργαστούμε στο σπίτι. Τα ρόπαλα, όμως, και ειδικά τα καλά, είναι φτιαγμένα από ξύλο. Είναι ζωντανά κι εύπλαστα. Μπορούν άνετα να μετατραπούν, από εμάς τους ίδιους, σε οτιδήποτε φανταζόμαστε. Αρκεί βέβαια να έχουμε τη γνώση και την υπομονή. Πάντως κανείς και τίποτε δεν μας περιορίζει. Μπορείς να μάθεις ξυλογλυπτική και να μετατρέψεις το ρόπαλό σου σ' οτιδήποτε: Ακόμα και σε σκελετό για το επόμενο ρομποτάκι σου!

Τα ρόπαλα υπάρχουν παντού. Μπορείς να αγοράσεις ένα ρόπαλο από το τοπικό κατάστημα εργαλείων (ή πρωτόγονων όπλων). Μπορείς να βρεις ένα ρόπαλο στην άκρη του δρόμου ή σε κάποιο μονοπάτι στο δάσος. Πάνω απ' όλα, όμως, μπορείς να κατασκευάσεις το δικό σου ρόπαλο! Μπορείς να το φτιάξεις από οποιοδήποτε υλικό και με οποιαδήποτε τεχνική. Για να είναι όμως καλό, θα πρέπει να χρησιμοποιήσεις μόνο άλλα ρόπαλα. Σωστά καταλάβατε. Τα ρόπαλα που αξίζουν κατασκευάζονται από άλλα ρόπαλα. Για την κατασκευή ενός καλού ροπάλου δεν χρησιμοποιούνται ποτέ εργαλεία από κάποια πολυεθνική. Ένα τέτοιο ρόπαλο θα άξιζε μόνο για προσάναμμα στο τζάκι! Θα ήταν π'τάχ, που λένε κι οι Κλίγκον...

~~~

Το 'χω παρατηρήσει κι εγώ. Είναι κάποιες φορές που ενδιαφερόμαστε περισσότερο για το αποτέλεσμα της εργασίας μας και λιγότερο για την ίδια τη διαδικασία. Για την ακρίβεια, μας ενδιαφέρει αρκετά και η διαδικασία, αλλά σκεφτόμαστε κυρίως το πώς θα απλοποιηθεί και θα γίνει ευκολότερη. Ακριβώς γι' αυτό, κάθε φορά επιλέγουμε τα κατάλληλα εργαλεία, χωρίς να επιτρέπουμε στη σκέψη μας να σκαλώνει. Το Free/Libre Open Source Software έχει αναρίθμητα προτερήματα. Θα μπορούσα να σας μιλάω για τα πολύτιμα διδάγματα και τα οφέλη που προσφέρει η χρήση του FLOSS για ατελείωτες ώρες. Κυριολεκτώ! Αυτό, όμως, πάντα κατά την ταπεινή γνώμη του πιστού αρχισυντάκτη, δεν σημαίνει ότι το FOSS συνιστά πάντα την καλύτερη επιλογή.

Προσοχή στον τυφλό φανατισμό. Οι μηχανικοί, οι προγραμματιστές, οι επιστήμονες γενικότερα αλλά κι όσοι προσπαθούν να κερδίζουν τον τίτλο του αληθινού χάκερ, έχουν ένα κοινό: Δεν φοράνε παρωπίδες, ούτε τις ανέχονται. Αυτό δεν σημαίνει ότι *δεν* έχουν σταθερές απόψεις. Σημαίνει, όμως, ότι δεν αντιμετωπίζουν ισοπεδωτικά τους πάντες και τα πάντα. Λαμβάνουν υπόψη τους τις ιδιαιτερότητες κάθε θέματος, καθώς και τις γενικότερες συνθήκες εντός των οποίων εκδηλώνεται καθετί. Με λίγα λόγια, προσπαθούν κάθε φορά να καταλήξουν στο πιο λογικό συμπέρασμα και στην πιο σωστή επιλογή.

Ακόμα και η ύπαρξη του FLOSS, αποτελεί κάτι το καταπληκτικό! Το ίδιο το FLOSS, όμως, δεν αποτελεί πάντα την καλύτερη επιλογή και δεν μπορεί να θεωρείται μονόδρομος. Τουλάχιστον όχι ακόμα. Μέχρι να συμβεί κάτι τέτοιο, η υιοθέτησή του για *κάθε* εργασία και σε *κάθε* περίσταση θυμίζει το φανατισμένο βάνδαλο-οπαδό, ο οποίος υποστηρίζει τυφλά την ομάδα του χωρίς κανένα λογικό επιχείρημα. Η ειρωνεία της υπόθεσης ξέρετε ποια είναι; Ο βάνδαλος-οπαδός κάνει τη μεγαλύτερη ζημιά στον σκοπό που (νομίζει ότι) υπηρετεί...

Spir@lEvolution


~~~





ουέλ μέλ

Αγαπητοί μου,

Παρατηρώ με χαρά ότι έχουμε πλέον μπει για τα καλά στην άνοιξη. Τα βράδια την έχει την ψυχρούλα του αλλά σε γενικές γραμμές τα πράγματα προχωρούν κατά το σχέδιο.

Σε λίγο θα έχουμε Πάσχα (την στιγμή που γράφονται αυτές εδώ οι γραμμές είμαστε ακόμα στον Μάρτιο, αλλά φέτος το Πάσχα δεν αργεί, διαβάζω), μετά είναι οι εκλογές, μετά η Πρωτομαγιά (ή μήπως the other way around;) -- πάντως με τούτο και με τ' άλλο, πριν προλάβουμε να το σκεφτούμε, θα είμαστε στο καλοκαίρι.

Αδημονώ, πρέπει να σας πω. Αδημονώ βασικά για λόγους χρώματος και φωτός.

Το καλοκαίρι ο ήλιος είναι ψηλά· σε καίει μεν αλλά δεν σε τυφλώνει. Θυμάμαι --πότε ήτανε;-- που είχα πάει στο Λονδίνο να επισκεφτώ τον ανιψιό και, αν και είχα πέσει σε καλό καιρό με κάτι υπέροχες λιακάδες, δεν είχα καταφέρει να το απολαύσω πραγματικά γιατί όποτε έβγαινα λίγο έξω για να με δει ο ήλιος αυτός έβρισκε την ευκαιρία να μου μπαίνει στα μάτια.

Τέλος πάντων, η κεντρική ιδέα είναι ότι αδημονώ.

Με έχει πιάσει η άνοιξη, αγαπητοί μου, και έχω αποφασίσει σήμερα να μη μιλήσω για το «Ελληνικό Ζήτημα». Άλλωστε είπα όσα ήθελα να πω και δεν μου αρέσει να επαναλαμβάνομαι.

Αλλά δεν μπορώ να αποφύγω να θίξω κάποια άλλα *συναφή* θέματα.

Κατ' αρχάς, θα το μάθατε βέβαια, πριν από μερικές εβδομάδες αποφάσισαν να ασχοληθούν με την πάρτη μας οι Anonymous. Δεν ξέρω αν το είδατε το βίντεο που ανέβασαν στο σάιτ του Υπουργείου Δικαιοσύνης. Εγώ το είδα. Πρέπει να το παραδεχτώ ότι είχε κάτι το θριαμβευτικό, σχεδόν αμείλικτο. Πήγα να χαρώ.

Ακούγοντας προσεκτικά αυτά που έλεγαν, πάντως, μου αμβλύνθηκε λίγο ο ενθουσιασμός. Τα ερωτήματά μου αυξήθηκαν τις επόμενες μέρες όταν είδα ένα δεύτερο βίντεο, πάλι των Anonymous, που κυκλοφόρησε με αφορμή τα επεισόδια του Φεβρουαρίου στην Αθήνα.

Πολύ απλά μας τα λένε αυτοί οι Anonymous. Για όλα φταίει, λέει, η Ελληνική Κυβέρνηση που δεν ακούει τον Λαό και τον αδικεί, γι αυτό και οι Anonymous, λέει, συμπαραστέκονται στον Λαό και θα ρίξουν την Κυβέρνηση, λέει. Όμορφα και απλά.

Μάλιστα επειδή έχουν επίγνωση του πόσο παγκοσμιοποιημένα είναι τα πράγματα σήμερα, οι Anonymous, λέει, απειλούν και την κυβέρνηση της Ιταλίας, της Ισπανίας, της Πορτογαλίας, της Μ.Βρετανίας -- με λίγα λόγια όλους!

Τι να σας κάνω, ανώνυμοι είναι, άσ' τους να λένε. Να, ακόμα κι εγώ, καλή ώρα, μπορώ να φτιάξω ένα βιντεάκι των Anonymous και να αρχίσω να κατηγορώ την Μαιρούλα επειδή ακόμα να μου επιστρέψει εκείνο το μνημοστικάκι που της είχα δώσει πέρσι (ή μήπως ήταν πρόπερσι;) ή την μητέρα

της Εξαδέλφης, που είναι Ινδουίστρια, επειδή ακριβώς είναι Ινδουίστρια και δεν είναι, ξέρω 'γω, κάτι άλλο πιο ελληνοπρεπές και εθνικό.

Αλλά αυτή η ευκολία με την οποία βγαίνουν όλοι και χρησιμοποιούν συλλογικούς όρους όπως π.χ. ο «Λαός», «οι Εργαζόμενοι», «η Εξουσία», «το Κεφάλαιο», πάντα μου στεκόταν στον λαιμό.

Πάρτε ας πούμε το «Λαός». Αγγλιστί «the People». Αυτόν τον όρο χρησιμοποιούν οι Anonymous: «The Greek People». Ωραίος όρος, δεν λέω, αλλά σε τι αναφέρεται;

Σας φαίνεται αυτονόητο, υποθέτω. Λαός είναι ο λαός, θα μου πείτε. Δεν υπάρχει και κανένα μυστήριο.

Εμ, δεν είναι εκεί το μυστήριο. Το μυστήριο αρχίζει όταν βλέπεις τον πάσα ένα να βγαίνει και να σου εξηγεί «τι θέλει ο λαός», χωρίς να μπορούν να συμφωνήσουν ούτε στα βασικά. «Ο Λαός θα αντισταθεί στη λαίλαπα της λιτότητας», σου λέει ο ένας. «Ο Λαός μένει σταθερά στην ευρωπαϊκή προοπτική της χώρας», σου λέει ο άλλος. «Ο Λαός δεν μπορεί να ανεχτεί άλλους μετανάστες», σου λέει ο τρίτος. «Ο Λαός αντιστέκεται στον εκφασισμό της κοινωνίας», σου λέει ο τέταρτος.

Ρε παιδιά, αποφασίστε επί τέλους! Αμάν. Το κεφάλι μας πάει να σπάσει. Ο Λαός αυτό, ο Λαός εκείνο. Τι θέλει επιτέλους ο Λαός; Βρείτε τα. Μοιραστείτε τα. Ποιόν Λαό θέλεις εσύ; Πάρ' τονα. Άσε όμως τον δικό μου κάτω. Και συ, εκεί στη γωνία, σταμάτα να λιμπίζεσαι το Λαό του πλησίον σου, 'ντάξει; Θα ήθελες να έρθουμε να σου πάρουμε τον δικό σου;

Και πάει λέγοντας.

Να μην τα πολυλογώ, πολύ στερεοτυπικά μας τα έλεγαν οι εν Ελλάδι Anonymous, και υπό την έννοια αυτή δεν μπορώ να πω ότι με εξέπληξε, αργότερα, όταν έμαθα ότι η αστυνομία μπούκαρε σε διάφορα σπίτια ανά την



επικράτεια (Τρίπολη, λέει, Αθήνα, Χαλκίδα, Ναύπλιο, Κατερίνη) και συνέλαβε διάφορα ελληνόψυχα τζιμάνια της Greek Hacking Scene, ως μέλη των Anonymous.

Δεν φημίζονται, δα, και για τη βαθύνοιά τους.

Οι συλληφθέντες αρνούνται φυσικά την εμπλοκή τους στο θέμα, και διαρρηγνύουν τα ιμάτιά τους για την αδικία που τους έλαχε.

Πλάκα πλάκα, μπορεί και να έχουν δίκιο, ξέρετε. Ελληνική Αστυνομία είναι αυτή, δεν μιλάμε για κανένα Σέρλοκ Χολμς. Πρώτη φορά θα είναι που ορμάνε μπούγιο και την πέφτουν σε όποιον βρουν; Τι, επειδή έχουν κομπιούτορες και ξέρουν, τάχα μου, από ηλεκτρονικό έγκλημα; Όποιον πάρει ο χάρος.

Μειδιώ, αγαπητοί μου. Μειδιώ.

Τώρα θα με κατηγορήσετε για εμπάθεια, το ξέρω. Σας διαβεβαιώ, ωστόσο, αγαπητοί μου, εμπάθεια ουδεμία! Εγώ με τα Όργανα της Τάξεως έχω τις καλύτερες σχέσεις. Με όλα τα Όργανα. Είτε κρατάνε κομπιούτορες, είτε το παίζουν μηχανόβιοι, είτε μπουζουριάζουν μετανάστες, είτε ραντίζουν τον Λαό (you see;) με δακρυγόνα, είτε συλλαμβάνουν φοροφυγάδες, είτε ρουφάνε φραπεδιές και γλυκοκοιτάνε τα πιπίνια. Εγώ τους συμπαθώ και νοιάζομαι για την υγεία τους.

Απλά λέω ότι μπορεί και να έχουν δίκιο (οι χάκερ). Υπάρχει, άλλωστε, και το τεκμήριο της αθωότητας. Να μην ξεχνιόμαστε.

Αυτό που με διασκέδασε ιδιαιτέρως, πάντως, ήταν κάτι που διάβασα για έναν από τους συλληφθέντες, τον γνωστό (σε άλλους) και ως Delirium. Ο πατέρας του --αντιγράφω-- «επισημαίνει πως ο γιός του δεν έχει σχέση με την επίθεση, τονίζοντας πως όταν κατάλαβε ότι ο γιός του ασχολείται με το χάκινγκ ειδοποίησε την αστυνομία.»

Τώρα πραγματικά, ελάτε στην θέση αυτού του δύσμοιρου πατέρα και πείτε μου. Τι μπορεί να νοιώθει ένας γονιός όταν καταλαβαίνει πως ο γιός του ασχολείται με το χάκινγκ; Πρώτα πρώτα: Πώς το καταλαβαίνει; Διαβάζει τα log;

Αλλά πείτε ότι το κατάλαβε· τώρα τι κάνει; Τον καλεί να το συζητήσουνε σαν άντρες; Κάνει δημόσια την αυτοκριτική του για τα λάθη του ως γονιού; Του προτείνει να πάνε μαζί σε κάποιον ειδικό να το κοιτάξουνε; Ή τον δίνει χωρίς περίσκεψη, χωρίς λύπη, χωρίς αιδώ, στο πρώτο Όργανο της Τάξης που θα συναντήσει μπροστά του;

Πραγματικά αναρωτιέμαι.

Διαβάζω πιο κάτω: «Φίλοι και συμμαθητές των τριών νεαρών που συνελήφθηκαν , που προέρχονται από οικογένειες με μικρή έως ασήμαντη οικονομική επιφάνεια, τους περιγράφουν ως αυθεντίες στο χάκινγκ και το κράκινγκ, επισημαίνοντας ότι ασχολούνται από πολύ μικρή ηλικία με τους υπολογιστές.»

Νοιώθω μια ζήλια, είμαι σίγουρος ότι με καταλαβαίνετε. Ζηλεύω και φθονώ γιατί εμένα ποτέ κανείς δεν με είπε αυθεντία ούτε στο χάκινγκ, ούτε στο κράκινγκ. Ούτε καν ο ανιψιός.

Βέβαια μάλλον είναι ότι δεν ασχολιόμουν από πολύ μικρή ηλικία με τους υπολογιστές. Αλλά πείτε μου, εγώ φταίω δηλαδή που δεν υπήρχαν υπολογιστές όταν ήμουν σε πολύ μικρή ηλικία;

Αδικία, φίλοι μου. Μεγάλη αδικία. Κάποιος πρέπει να κάνει κάτι. Τι περιμένουν οι Anonymous;

Σας ασπάζομαι,
Θείος Ακάκιος

Εσείς τι δικαιολογία έχετε που η διαφήμισή σας απουσιάζει από εδώ;

Τολμήστε να μας την πείτε από το talk2us@deltahacker.gr. Όπως και να 'χει θα επικοινωνήσουμε κι εμείς μαζί σας - ή έστω με τον ανταγωνιστή σας ;)

Skill: Intermediate

Tags: VPN, OpenVPN, encryption, privacy, tunneling, VPS, Hulu, Netflix

Φως στην άκρη του τούνελ!

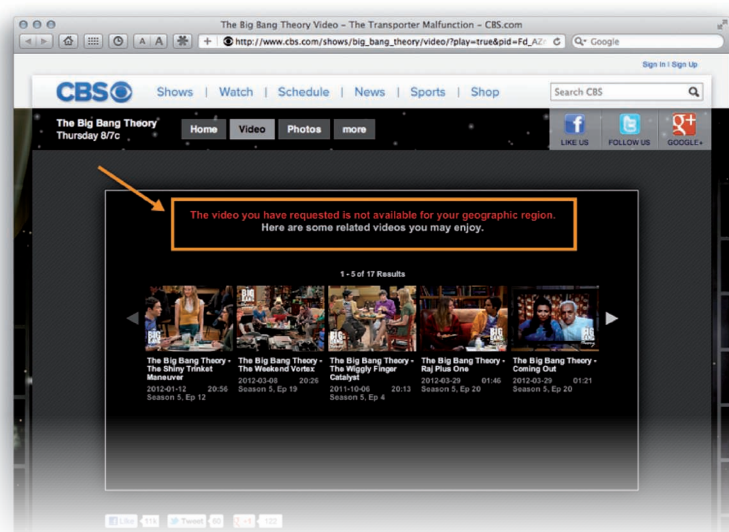
by subZraw

Μην ανησυχείτε, ο τίλος δεν είναι πολιτικάντικος, ούτε εμπνευσμένος από την προεκλογική περίοδο. Αντιθέτως, σχεδόν κυριολεκτούμε: Σε λίγο θα σας δείξουμε πώς ν' ανοίγετε δικτυακά τούνελ από τον υπολογιστή σας προς κάποιον άλλον, μέσα από τα οποία με χαρά θα θέλετε να ταξιδεύετε! Εκτός από την απόλυτη ασφάλεια που παρέχουν, στις εξόδους τους θα σας περιμένουν ένα σωρό ενδιαφέροντα πράγματα που μέχρι χθες ακούγατε ότι υπάρχουν, αλλά στα οποία δεν ήταν καθόλου εύκολο να έχετε πρόσβαση.

Ο λόγος γίνεται για τα δίκτυα VPN ή, αν θέλουμε να αποφύγουμε τον πλεονασμό, απλά για τα VPNs (Virtual Private Networks). Ένας υπολογιστής που συμμετέχει σε κάποιο VPN έχει ασφαλέστατη πρόσβαση σε πόρους και υπηρεσίες ενός απομακρυσμένου, τοπικού δικτύου, ασχέτως του πού βρίσκεται ο ίδιος κι ανεξαρτήτως του αν χρησιμοποιεί ασφαλή τρόπο σύνδεσης ή όχι. Για παράδειγμα, χάρη στην τεχνολογία των VPN ένας εργαζόμενος αποκτά πρόσβαση στο εταιρικό δίκτυο από το σπίτι του, από μια καφετέρια, κάποιο αεροδρόμιο, ένα ξενοδοχείο κ.ο.κ. Ομοίως, ένας χρήστης που παίρνει στα σοβαρά το θέμα της ασφάλειας, έχει στήσει έναν VPN server κι έτσι μπορεί να συνδέεται στον υπολογιστή ή στους υπολογιστές του σπιτιού του ακόμη κι από το δημόσιο, επισφαλές δίκτυο της πλατείας.

Τώρα, πέρα από την ασφαλή, κρυπτογραφημένη πρόσβαση στους πόρους ενός απομακρυσμένου LAN, ο χρήστης του VPN έχει τη δυνατότητα να περνά *όλη* τη δικτυακή κίνηση του υπολογιστή του *μέσα* από το VPN. Έτσι, το web browsing γίνεται μέσω ενός ασφαλούς καναλιού: Πρόκειται για το ίδιο κανάλι που ξεκινά από τον λεγόμενο VPN client και φτάνει έως τον απομακρυσμένο VPN server. Κανένα πρόβλημα, λοιπόν, αν κάπου στην πλατεία βρίσκεται ένας ανήσυχος χρήστης με το Pineapple του, ο οποίος διασκεδάζει υποκλέπτοντας τις συνδέσεις άλλων χρηστών της περιοχής: Ακόμη και να υποκλέψει το traffic του VPN client, από τη στιγμή που είναι κρυπτογραφημένο θα του είναι εντελώς άχρηστο. (Για το Pineapple, τη λογική λειτουργίας του και το τι μπορεί να κάνει, δείτε το σχετικό άρθρο που αρχίζει από τη σελίδα 22 του deltaHacker 005.) Φυσικά, δεν είναι απαραίτητο να 'χει κάποιος Pineapple, προκειμένου να καταλύσει την ασφάλεια των απρόσεκτων χρηστών. Για παράδειγμα, ένα laptop με το BackTrack Linux είναι ό,τι ακριβώς χρειάζεται για να υλοποιηθούν ένα σωρό ενδιαφέροντα MiTM attacks. Αλλά ας μην ξεφεύγουμε.

Τα VPN, λοιπόν, προσφέρουν μεγάλη ευκολία κι άριστη ασφάλεια. Την ίδια στιγμή, υπό προϋποθέσεις μάς "ανοίγουν" υπηρεσίες στις οποίες δεν θα ήταν δυνατόν να έχουμε πρόσβαση, τουλάχιστον όχι χωρίς να κάναμε πρώτα ένα υπερατλαντικό ταξίδι. Ενδεχομένως να 'χετε ακούσει για το



Πρώτη απόπειρα να δούμε το πιο πρόσφατο επεισόδιο του The Big Bang Theory, από το cbs.com. Μιας και η υπηρεσία είναι διαθέσιμη μόνο για αμερικανούς, τρώμε πόρτα :S

Hulu, ένα site όπου ο ενδιαφερόμενος παρακολουθεί επεισόδια αμερικάνικων τηλεοπτικών σειρών, μάλιστα ασχέτως αν μιλάμε για παλιές σειρές ή καινούργιες, που προβάλλονται τώρα. Το θέμα με το Hulu είναι ότι επιτρέπει την πρόσβαση *μόνο* σε αμερικανούς.

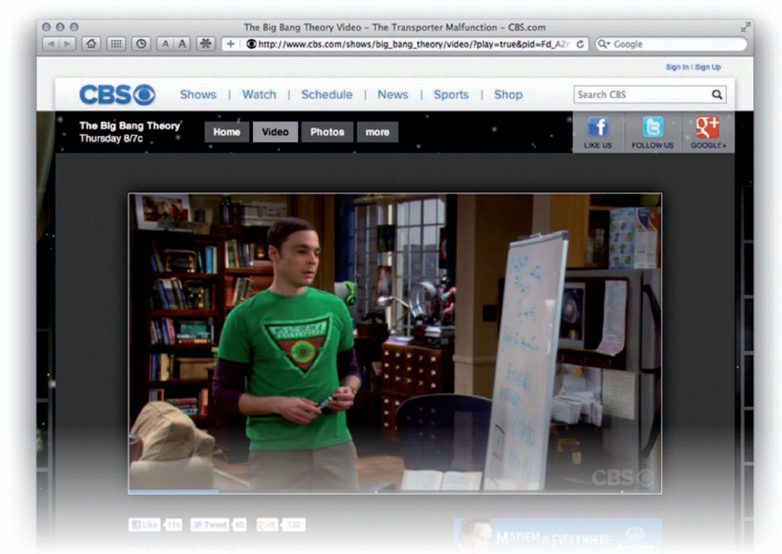
Τώρα, κάποιος εκτός αμερικής που θα 'θελε να 'χει πρόσβαση στο Hulu και σ' άλλες παρόμοιες υπηρεσίες, ως λύση θα μπορούσε να σκεφτεί τη χρήση ενός ανοικτού proxy server που φυσικά θα βρίσκεται σε αμερικάνικο έδαφος. Όμως με την προσέγγιση των proxy υπάρχουν δύο σοβαρά προβλήματα. Το ένα έχει να κάνει με τη χαμηλή ταχύτητα πρόσβασης που παρέχουν, η οποία είναι απλά ακατάλληλη για εφαρμογές streaming (σκεφτείτε μόνο πόσοι τους χρησιμοποιούν). Το άλλο πρόβλημα είναι ότι, πολύ συχνά, οι proxies μπαίνουν σε μαύρες λίστες από τους παρόχους περιεχομένου, οι οποίοι απλά δεν δέχονται συνδέσεις που προέρχονται από μηχανήματα εντός τέτοιων λιστών.

Τελικά, όπως διαπιστώσαμε κι εμείς στην πράξη, η μία και μοναδική λύση για την πρόσβαση σε υπηρεσίες σαν το Hulu ή το Netflix, είναι να 'χουμε τον δικό μας VPN server κάπου στην Αμερική. Έτσι, κάθε φορά που συνδεόμαστε σ' αυτόν βγαίνουμε στο ίντερνετ με αμερικάνικο IP και ταυτόχρονα δεν είμαστε black listed. Και πώς όμως θ' αποκτήσουμε VPN server στην Αμερική; Μα, αγοράζοντας ένα καλό, αξιόπιστο και φθηνό VPS (Virtual Private Server) με Linux, το οποίο θα τρέχει το κατάλληλο VPN software!

Προαπαιτούμενα για τη συνέχεια

Στο υπόλοιπο του παρόντος δείχνουμε πώς εγκαθιστούμε και ρυθμίζουμε το OpenVPN, μια εφαρμογή ανοικτού λογισμικού που υλοποιεί έναν VPN server, ενώ μπορεί να παίζει και το ρόλο VPN client.

Το λειτουργικό σύστημα στο οποίο θα δουλέψουμε είναι το Ubuntu Server (11.10, 32bit), με την υπηρεσία SSH ενεργοποιημένη. Η δε υπολογιστική πλατφόρμα κάλλιστα μπορεί να 'ναι ένας αληθινός υπολογιστής, μια εικονική μηχανή (virtual machine, VM) ή κάποιο VPS που έχουμε στην κατοχή



Δεύτερη απόπειρα να δούμε το πιο πρόσφατο επεισόδιο του The Big Bang Theory, πάλι από το cbs.com. Τώρα βγαίνουμε στο Internet μέσω OpenVPN server, ο οποίος τρέχει σε VPS που έχουμε σ' ένα data center κάπου στο New Jersey. Αυτή τη φορά, λοιπόν, βλέπουμε ότι... it's showtime! Αλλά για μισό λεπτό, αυτό είναι το όνομα άλλου αμερικάνικου τηλεοπτικού δικτύου :D

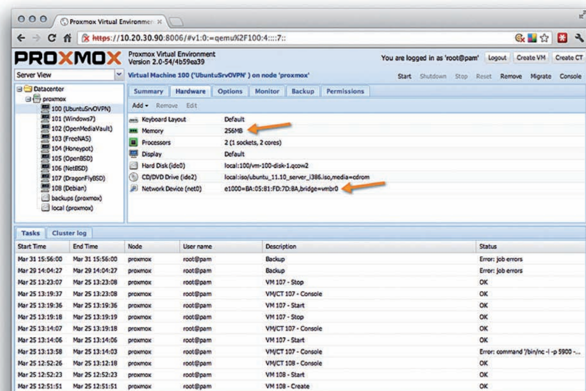
Αν στήνουμε τον OpenVPN server σε VM, τότε καλό είναι να φροντίσουμε ώστε η εικονική κάρτα Ethernet να 'ναι στο λεγόμενο bridged mode. Εξάλλου, τα 256MB μνήμης RAM είναι υπεραρκετά για ένα VM που ως μοναδική του αποστολή έχει να τρέχει το OpenVPN. Και μην πει κάποιος ότι είναι πολλά: O Firefox στην καθησιά του γύρω στο μισό giga RAM θέλει!

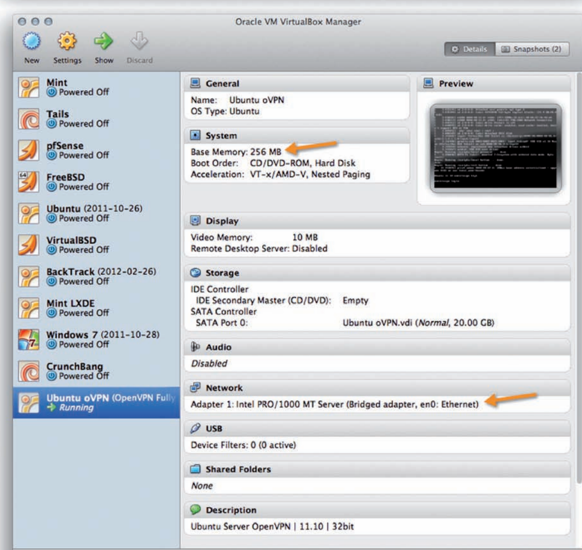
μας. Αν η περίπτωση του VPS σάς φαίνεται ελκυστική αλλά δεν ξέρετε από πού ν' αρχίσετε ώστε ν' αποκτήσετε το δικό σας, αξίζει να διαβάσετε το σχετικό άρθρο που δημοσιεύεται στο deltaHacker 005 (βλ. σελ. 44).

Στις περιπτώσεις που δουλεύουμε σε αληθινό ή εικονικό μηχάνημα πίσω από κάποιον (ADSL) router, εννοείται ότι πρέπει να 'μαστε σε θέση να ανοίξουμε το κατάλληλο port που θέλει το OpenVPN ή/και να δημιουργήσουμε τον απαραίτητο κανόνα port forwarding, από τον router προς το Ubuntu box. Το λειτουργικό σύστημα εξάλλου προτείνεται να 'χει στατικό IP. Εξ ορισμού, το Ubuntu Server μετά την εγκατάσταση έχει δυναμικό. Το πώς μπορείτε να του δώσετε στατικό περιγράφεται στη σελίδα 81 του deltaHacker 006. Εναλλακτικά, δείτε το σχετικό αρθράκι στο Parabing:

```
http://parabing.com/?p=728
```

Το IP της δικής μας εγκατάστασης, για το δικό μας τοπικό δίκτυο, ήταν το 10.20.30.235. Στη δική σας περίπτωση κατά πάσα πιθανότητα θα 'ναι διαφορετικό, π.χ., της μορφής 10.0.0.*, 192.168.0.*, 192.168.1.* κ.ο.κ.





Στην αμέσως επόμενη ενότητα του άρθρου περνάμε στη δράση, δείχνοντας αναλυτικά και βήμα προς βήμα πώς ρυθμίζουμε τον OpenVPN server σ' ένα KVM-based VM με Ubuntu Server 11.10, το οποίο τρέχει σε PC με το Proxmox VE (www.proxmox.com/products/proxmox-ve). Φυσικά, εντελώς ανάλογα δουλεύουμε σε περιβάλλον VirtualBox, VMware, VPS ή αληθινού υπολογιστή. Ειδικά στις περιπτώσεις των Proxmox, Virtualbox και VMware, η κάρτα Ethernet πρέπει να 'ναι σε bridged mode -- μ' άλλα λόγια να "βλέπει" απευθείας τον router του LAN.

Εκτός από τη ρύθμιση του server θα δείξουμε φυσικά και πώς συνδεόμαστε σ' αυτόν από άλλον υπολογιστή ή laptop, μάλιστα ασχέτως αν τρέχει Windows, Linux ή OS X. Προς το τέλος του άρθρου θ' αναφερθούμε στη δική μας εμπειρία χρήσης του OpenVPN ενός VPS, το οποίο εδώ και πολύ καιρό είχαμε στην Αγγλία αλλά ζητήσαμε και μας το μετέφεραν στην Αμερική.

Ρύθμιση του server

Για τις ανάγκες της παρουσίασης, χωρίζουμε την όλη διαδικασία σε βήματα.

Βήμα 1. Κάνουμε login στο Ubuntu Server, από την κονσόλα ή μέσω SSH, κι αμέσως αλλάζουμε στον

λογαριασμό του root:

```
sub0@oneiricvpn:~$ sudo su
[sudo] password for sub0: πληκτρολογούμε το password του χρήστη μας
root@oneiricvpn:/home/sub0#
```

Δεν είναι απαραίτητο, καλού κακού όμως κάνουμε ένα φρεσκάρισμα στο λειτουργικό:

```
root@oneiricvpn:/home/sub0# apt-get update
root@oneiricvpn:/home/sub0# apt-get -y dist-upgrade
```

Αν πρόκειται για καινούργια εγκατάσταση του Ubuntu Server, τότε μετά την ολοκλήρωση του dist-upgrade θα ‘χει αναβαθμιστεί κι ο πυρήνας. Καλό είναι λοιπόν να κάνουμε μια επανεκκίνηση.

Βήμα 2. Συνεχίζουμε --πάντα από τον λογαριασμό του root-- με την εγκατάσταση του OpenVPN:

```
root@oneiricvpn:/home/sub0# apt-get -y install openvpn
```

Βήμα 3. Για την εγκαθίδρυση κρυπτογραφημένων τούνελ μεταξύ clients και OpenVPS server δημιουργούνται ορισμένα πιστοποιητικά (certificates), καθώς και κάποια σετ ιδιωτικών/δημοσίων κλειδιών (private και public keys). Αν δεν έχετε ακούσει ξανά για όλα αυτά ή δεν είστε σίγουροι για το πώς δουλεύουν στη συγκεκριμένη περίπτωση, διαβάστε το άρθρο που έχουμε στο

http://deltahacker.gr/?p=5058

Το στήσιμο της υποδομής κρυπτογράφησης για το OpenVPN γίνεται με τη βοήθεια συγκεκριμένων scripts, τα οποία μετά την εγκατάσταση του πακέτου openvpn βρίσκονται στον κατάλογο

```
/usr/share/doc/openvpn/examples/easy-rsa/2.0
```

Όλος ο κατάλογος 2.0 πρέπει να βρίσκεται κάτω από τον

```
/etc/openvpn/easy-rsa
```

Με τις ακόλουθες εντολές φροντίζουμε γι’ αυτό:

```
root@oneiricvpn:/home/sub0# mkdir /etc/openvpn/easy-rsa
root@oneiricvpn:/home/sub0# cd /etc/openvpn/easy-rsa
root@oneiricvpn:/etc/openvpn/easy-rsa# cp -R \
/usr/share/doc/openvpn/examples/easy-rsa/2.0 .
```

Βήμα 4. Για τη συνέχεια, οφείλουμε να τροποποιήσουμε τις τιμές ορισμένων μεταβλητών στο αρχείο vars, μέσα στον κατάλογο 2.0. Ένας εύχρηστος text editor γι’ αυτόν το σκοπό είναι το nano:

```
root@oneiricvpn:/etc/openvpn/easy-rsa# cd 2.0
root@oneiricvpn:/etc/openvpn/easy-rsa/2.0# nano vars
```

Δείτε από το δικό μας παράδειγμα ποιες μεταβλητές πρέπει ν’ αλλάξουν και πράξτε αναλόγως για το δικό σας αρχείο vars (πηγαίνετε προς το τέλος του):

```
export KEY_COUNTRY="GR"
export KEY_PROVINCE="CentralMacedonia"
```

```
export KEY_CITY="Thessaloniki"
export KEY_ORG="deltaHacker"
export KEY_EMAIL="shadow@uovpn.deltahacker.gr"
```

Αποθηκεύστε τις αλλαγές ([CTRL+O], [Enter]) κι εγκαταλείψτε το nano ([CTRL+X]).

Βήμα 5. Παραμένοντας μέσα στον κατάλογο /etc/openvpn/easy-rsa/2.0 δίνουμε, διαδοχικά:

```
root@oneiricvpn:/etc/openvpn/easy-rsa/2.0# source ./vars
NOTE: If you run ./clean-all, I will be doing a rm -rf on /etc/
openvpn/easy-rsa/2.0/keys
root@oneiricvpn:/etc/openvpn/easy-rsa/2.0# sh ./clean-all
root@oneiricvpn:/etc/openvpn/easy-rsa/2.0# sh ./build-ca
Generating a 1024 bit RSA private key
[...]
-----
Country Name (2 letter code) [GR]: [Enter]
State or Province Name (full name) [CentralMacedonia]: [Enter]
Locality Name (eg, city) [Thessaloniki]: [Enter]
Organization Name (eg, company) [deltaHacker]: [Enter]
Organizational Unit Name (eg, section) []: [Enter]
Common Name (eg, your name or your server's hostname)
[deltaHacker CA]: [Enter]
Name []: [Enter]
Email Address [shadow@uovpn.deltahacker.gr]: [Enter]
```

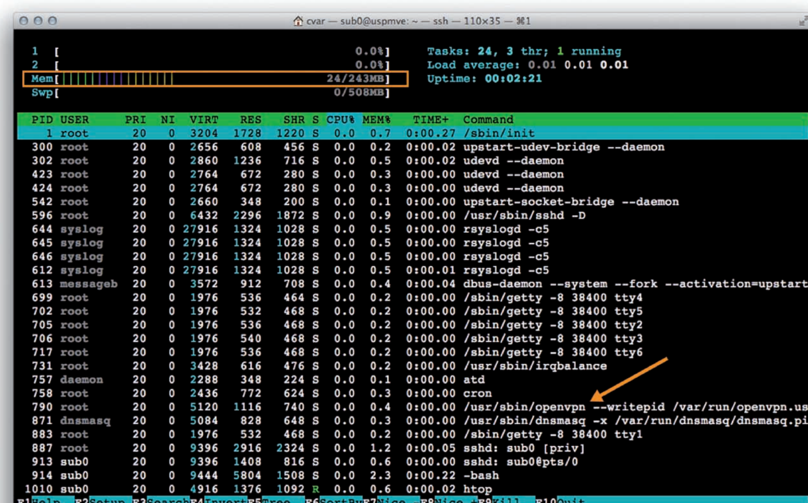
Επειδή πριν λίγο τροποποιήσαμε το αρχείο vars, σε όλες τις ερωτήσεις του build-ca αφήσαμε τις προεπιλογές πατώντας, απλά, το [Enter]. Μόλις δημιουργήσαμε αυτό που ονομάζεται Αρχή Πιστοποίησης (Certificate Authority, CA). Αναλυτικότερα...

- με την εντολή source ./vars αρχικοποιήσαμε όλες τις μεταβλητές περιβάλλοντος που χρησιμοποιούν τα σκριπτάκια clean-all και build-ca
- με την sh ./clean-all φροντίσαμε για την ύπαρξη του υποκαταλόγου keys, ο οποίος θα πρέπει να 'ναι προσπελάσιμος *μόνον* από τον χρήστη root
- με την sh ./build-ca φτιάξαμε το ιδιωτικό κλειδί (private key) της CA (αρχείο keys/ca.key), καθώς κι ένα πιστοποιητικό (certificate, αρχείο keys/ca.crt) που χρησιμοποιεί τόσο ο server, όσο και οι clients.

Το ιδιωτικό κλειδί οφείλουμε να το προστατεύουμε ως κόρη οφθαλμού και περιττό να πούμε ότι δεν πρέπει να διαρρεύσει.

Βήμα 6. Είναι ώρα να φτιάξουμε το ιδιωτικό κλειδί του OpenVPN server (διαφορετικό από το ιδιωτικό της CA). Πληκτρολογούμε την ακόλουθη εντολή (στη θέση του uovpn, που είναι το hostname του VM που δουλεύαμε, μπορείτε να βάλετε κάποιο άλλο όνομα):

```
root@oneiricvpn:/etc/openvpn/easy-rsa/2.0# sh ./build-key-server
uovpn
Generating a 1024 bit RSA private key
[...]
-----
Country Name (2 letter code) [GR]: [Enter]
State or Province Name (full name) [CentralMacedonia]: [Enter]
Locality Name (eg, city) [Thessaloniki]: [Enter]
Organization Name (eg, company) [deltaHacker]: [Enter]
Organizational Unit Name (eg, section) []:[Enter]
Common Name (eg, your name or your server's hostname) [uovpn]:
[Enter]
Name []: [Enter]
Email Address [shadow@uovpn.deltahacker.gr]: [Enter]
Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:[Enter]
An optional company name []:[Enter]
Using configuration from /etc/openvpn/easy-rsa/2.0/openssl.cnf
Check that the request matches the signature
Signature ok
The Subject's Distinguished Name is as follows
countryName           :PRINTABLE:'GR'
stateOrProvinceName   :PRINTABLE:'CentralMacedonia'
localityName          :PRINTABLE:'Thessaloniki'
organizationName      :PRINTABLE:'deltaHacker
commonName            :PRINTABLE:'uovpn'
emailAddress          :IA5STRING:'shadow@uovpn.deltahacker.gr'
Certificate is to be certified until Mar 28 04:27:14 2022 GMT
(3650 days)
```



Είδατε που σας το λέγαμε; Οι διεργασίες του VM μαζί με τον OpenVPN server καταναλώνουν συνολικά γύρω στα 24MB μνήμης RAM. Τελικά, αυτά τα 256MB που δώσαμε στην εικονική μηχανή μάλλον πολλά είναι...

```
Sign the certificate? [y/n]:y
1 out of 1 certificate requests certified, commit? [y/n]y
Write out database with 1 new entries
Data Base Updated
```

Σε όλες τις ερωτήσεις μπορείτε ν' αφήσετε τις προεπιλογές.

Βήμα 7. Η ανταλλαγή κλειδιών μεταξύ client και server αλλά και η διαδικασία της ταυτοποίησης (authentication), ρυθμίζονται από τις λεγόμενες παραμέτρους Diffie Hellman (δείτε το www.rsa.com/rsalabs/node.asp?id=2248 για μια αναλυτική επεξήγηση). Η δημιουργία των παραμέτρων επιτυγχάνεται πληκτρολογώντας:

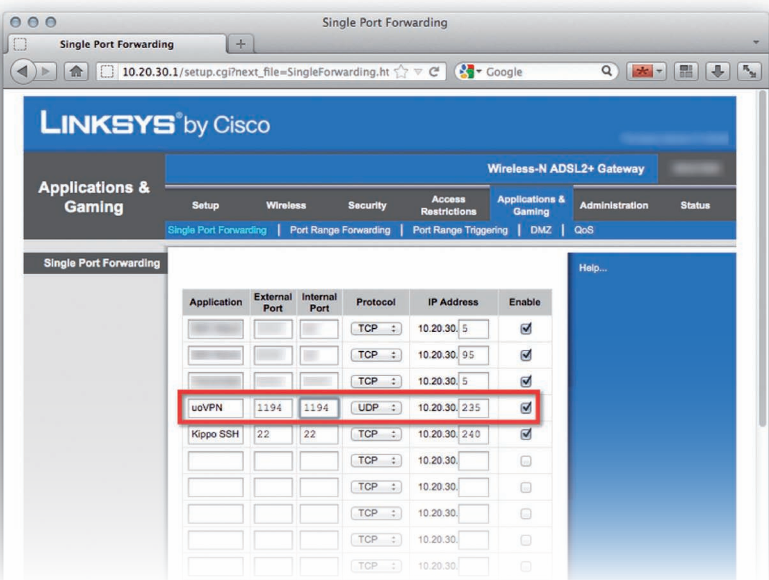
```
root@oneiricvpn:/etc/openvpn/easy-rsa/2.0# ./build-dh
Generating DH parameters, 1024 bit long safe prime, generator 2
This is going to take a long time
[...]
```

Η διαδικασία δεν ολοκληρώνεται ακαριαία και στη χειρότερη περίπτωση ενδέχεται να διαρκέσει μερικά λεπτά.

Βήμα 8. Τα πιστοποιητικά, τα κλειδιά και το αρχείο με τις παραμέτρους Diffie Hellman που φτιάξαμε στα προηγούμενα βήματα, βρίσκονται όλα τους μέσα στον κατάλογο /etc/openvpn/easy-rsa/2.0/keys. Πρόκειται για πέντε στο πλήθος αρχεία και για τη δική μας περίπτωση ήταν τα ακόλουθα:

1. ca.crt: το πιστοποιητικό της Αρχής Πιστοποίησης
2. ca.key: το ιδιωτικό κλειδί της Αρχής Πιστοποίησης
3. uovpn.crt: το πιστοποιητικό του OpenVPN server
4. uovpn.key: το ιδιωτικό κλειδί του OpenVPN server
5. dh1024.pem: το αρχείο παραμέτρων Diffie Hellman

Αν το OpenVPN τρέχει σε box ή VM πίσω από κάποιον οικιακό router (το πιθανότερο), τότε εννοείται ότι στον τελευταίο πρέπει να ορίσουμε τον κατάλληλο κανόνα port forwarding.



(Τα 3 και 4 πιθανώς θα έχουν διαφορετικό όνομα στη δική σας περίπτωση -- για την ακρίβεια αυτό που δώσατε τρέχοντας το σκριπτάκι build-key-server.) Τα πέντε αυτά αρχεία πρέπει να τα αντιγράψουμε στον κατάλογο /etc/openvpn:

```
root@oneiricvpn:/etc/openvpn/easy-rsa/2.0# cd keys
root@oneiricvpn:/etc/openvpn/easy-rsa/2.0/keys# cp ca.crt ca.key \
uovpn.crt uovpn.key dh1024.pem /etc/openvpn
```

Βήμα 9. Σε πολύ λίγο θα έχουμε τελειώσει με το στήσιμο του OpenVPN server. Μένει μόνο να φτιάξουμε ένα βασικό αρχείο ρυθμίσεων, το οποίο θα βάλουμε στον κατάλογο /etc/openvpn. Το αρχείο αυτό επιτρέπεται να 'χει όποιο όνομα θέλουμε, η κατάληξή του όμως πρέπει να 'ναι .conf. Υπάρχει ένα υπόδειγμα αρχείου ρυθμίσεων, συμπιεσμένο μέσα στον κατάλογο /usr/share/doc/openvpn/examples/sample-config-files. Είναι ακριβώς ό,τι χρειαζόμαστε:

```
root@oneiricvpn:/etc/openvpn/easy-rsa/2.0/keys# cd /etc/openvpn
root@oneiricvpn:/etc/openvpn# cp \
/usr/share/doc/openvpn/examples/sample-config-files/server.conf.
gz .
root@oneiricvpn:/etc/openvpn# gunzip -d server.conf.gz
```

Όπως είναι το αρχείο server.conf δεν χρειάζεται κάποια αλλαγή, εκτός ίσως από τις δύο ακόλουθες γραμμές:

```
cert server.crt
key server.key
```

Αναλόγως του ονόματος που έχετε δώσει στο πιστοποιητικό και στο ιδιωτικό κλειδί του OpenVPN server, τροποποιήστε όπως χρειάζεται. Εμείς, π.χ., τις κάναμε έτσι:

```
cert uovpn.crt
key uovpn.key
```

Βήμα 10. Η ενεργοποίηση του OpenVPN server επιτυγχάνεται πληκτρολογώντας

```
root@oneiricvpn:/etc/openvpn# service openvpn start
* Starting virtual private network daemon(s)...
*   Autostarting VPN 'server'                    [ OK ]
```

Μια χαρά όλα :D

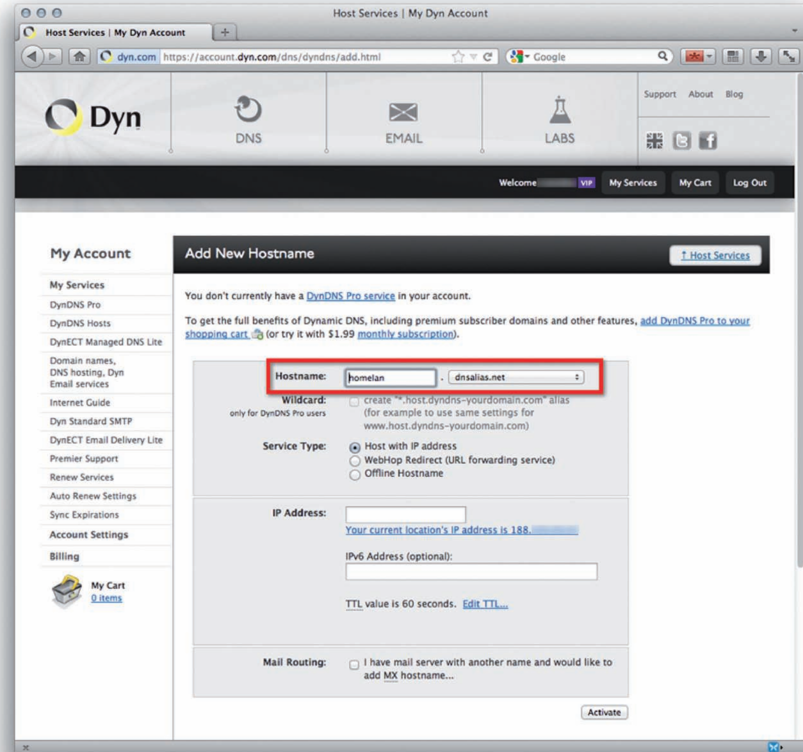
Ρύθμιση των clients και σύνδεση στο VPN

Ας πούμε ότι θέλουμε να συνδεόμαστε στον OpenVPN server από το laptop μας -- αυτό εξάλλου είν' ένα πολύ συνηθισμένο σενάριο. Πρέπει να δημιουργήσουμε ένα ιδιωτικό κλειδί κι ένα πιστοποιητικό για το laptop. Το σκριπτάκι που χρειαζόμαστε ονομάζεται build-key και βρίσκεται μέσα στον κατάλογο /etc/openvpn/easy-rsa/2.0:

```
root@oneiricvpn:/etc/openvpn# cd easy-rsa/2.0
root@oneiricvpn:/etc/openvpn/easy-rsa/2.0# sh ./build-key laptop
Generating a 1024 bit RSA private key
[...]
```

```
-----
Country Name (2 letter code) [GR]: [Enter]
State or Province Name (full name) [CentralMacedonia]: [Enter]
Locality Name (eg, city) [Thessaloniki]: [Enter]
Organization Name (eg, company) [deltaHacker]: [Enter]
Organizational Unit Name (eg, section) []:[Enter]
Common Name (eg, your name or your server's hostname) [laptop]:
[Enter]
Name []: [Enter]
Email Address [shadow@uovpn.deltahacker.gr]: [Enter]
Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []: [Enter]
An optional company name []: [Enter]
Using configuration from /etc/openvpn/easy-rsa/2.0/openssl.cnf
Check that the request matches the signature
Signature ok
The Subject's Distinguished Name is as follows
countryName           :PRINTABLE:'GR'
stateOrProvinceName   :PRINTABLE:'CentralMacedonia'
localityName          :PRINTABLE:'Thessaloniki'
organizationName      :PRINTABLE:'deltaHacker'
commonName            :PRINTABLE:'laptop'
emailAddress          :IA5STRING:'shadow@uovpn.deltahacker.gr'
Certificate is to be certified until Mar 28 14:35:48 2022 GMT
(3650 days)
Sign the certificate? [y/n]:y
1 out of 1 certificate requests certified, commit? [y/n]y
Write out database with 1 new entries
Data Base Updated
```

Όπως βλέπετε στο παράδειγμά μας, εμείς επιλέξαμε το καθόλου πρωτότυπο όνομα “laptop” (μπορείτε να βάλετε όποιο όνομα θέλετε), ενώ στις ερωτήσεις του build-key αφήσαμε τις προεπιλογές. Το script έφτιαξε ένα πιστοποιητικό (laptop.crt) κι ένα ιδιωτικό κλειδί (laptop.key) για τον client, και τ’ άφησε ασφαλή μέσα στον κατάλογο keys. Τώρα, τα αρχεία ca.crt --και ειδικά τα laptop.crt, laptop.key--, πρέπει να μεταφερθούν με απόλυτη ασφάλεια στον client (στο παράδειγμά μας, εκείνο το laptop). Αυτό μπορεί να γίνει με διάφορους τρόπους, όπως, π.χ., χειροκίνητα μ’ ένα USB stick ή δικτυακά με SFTP ή secure copy (αρκεί στο Ubuntu να ‘ναι ενεργοποιημένο το SSH). Με κάποιον τρόπο, λοιπόν, υποθέτουμε ότι τα αρχεία ca.crt, laptop.crt και laptop.key έχουν μεταφερθεί στο laptop. Καλό είναι μάλιστα να βρίσκονται όλα τους οργανωμένα μέσα σ’ έναν κατάλογο, στον οποίο θα ‘χει πρόσβαση μόνον ο χρήστης του απομακρυσμένου OpenVPN server. Σ’ αυτόν τον κατάλογο μένει να μεταφέρουμε κι ένα κατάλληλα διαμορφωμένο αρχείο ρυθμίσεων. Ένα υπόδειγμα υπάρχει στο /usr/share/doc/openvpn/examples/sample-config-files:



*Για άλλη μια φορά, αν ο OpenVPN server λειτουργεί σε οικιακό περιβάλλον, τότε το δημόσιο IP του router μπροστά του είναι μάλλον δυναμικό. Αυτό θα μπορούσε ν' αποτελεί πρόβλημα για τις συνδέσεις μας στον server, αν *δεν* υπήρχαν δωρεάν υπηρεσίες αντιστοίχισης domain name σε δυναμικά IP. Αυτή που χρησιμοποιούμε εμείς για χρόνια παρέχεται από τη Dyn (`http://dyn.com`). Άλλη γνωστή υπηρεσία της κατηγορίας είναι εκείνη που παρέχεται από το `http://no-ip.com`.*

```
root@oneiricvpn:/etc/openvpn/easy-rsa/2.0# cp \
/usr/share/doc/openvpn/examples/sample-config-files/client.conf ~/
root@oneiricvpn:/etc/openvpn/easy-rsa/2.0# cd
root@oneiricvpn:~#
```

Στο αρχειάκι client.conf πρέπει να κάνουμε δυο-τρεις αλλαγές. Το ανοίγουμε με το nano κι εντοπίζουμε τη γραμμή

```
remote my-server-1 1194
```

Στη θέση του my-server-1 πρέπει να βάλουμε το δημόσιο IP που έχει ο router μπροστά από τον OpenVPN server. (Για να το βρούμε, αρκεί να είμαστε εντός LAN και να επισκεφτούμε ένα site σαν το `http://checkip.dyndns.org`.) Βεβαίως, επειδή το IP αυτό κατά πάσα πιθανότητα είναι δυναμικό κι ενδέχεται ν' αλλάξει ανά πάσα στιγμή, είναι προτιμότερο να καταφύγουμε σε μια δωρεάν υπηρεσία σαν αυτή του `http://dyn.com` ή του `http://no-ip.com`, ώστε ν' αντιστοιχίσουμε ένα domain name στο IP που ουσιαστικά παίρνουμε από τον ISP μας. Έτσι, αντί να βάλουμε το δημόσιο IP του router στη θέση του my-server-1, μπορούμε να βάλουμε ένα πλήρες domain σαν το homelan.dnsalias.net και η παραπάνω γραμμή να γίνει

```
remote homelan.dnsalias.net 1194
```

Αυτό το 1194, στα δεξιά, είναι το default port που χρησιμοποιεί ο OpenVPN server. Μόνος λόγος για να τ' αλλάξουμε είναι αν έχουμε ήδη ορίσει κάποιο άλλο port στο config file του server. Όπως και να 'χει, στον router μας πρέπει να φροντίσουμε για τον κατάλληλο κανόνα port forwarding,

ώστε τα εισερχόμενα πακέτα UDP προς το port 1194 του router να ανακατευθύνονται προς το port 1194 του box στο οποίο τρέχει ο OpenVPN server. Δύο άλλες γραμμές του client.conf που ίσως χρειαστεί να πειράξουμε είναι οι ακόλουθες:

```
cert client.crt
key client.key
```

Στο παράδειγμά μας, το πιστοποιητικό και το ιδιωτικό κλειδί του client είναι τα αρχεία laptop.crt και laptop.key αντίστοιχα. Έτσι, οι δύο παραπάνω γραμμές γίνονται

```
cert laptop.crt
key laptop.key
```

Αποθηκεύουμε τις αλλαγές στο client.conf και μεταφέρουμε το αρχείο στο laptop -- κατά προτίμηση στον κατάλογο που περιλαμβάνει τα αρχεία ca.crt, laptop.crt και laptop.key.

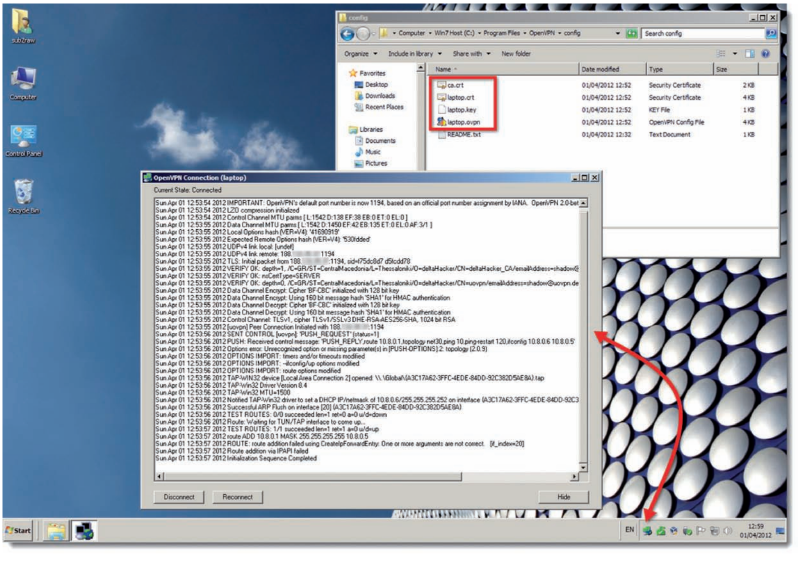
Είναι ώρα να δούμε πώς συνδεόμαστε στον OpenVPN server. Παίρνουμε τρεις περιπτώσεις λειτουργικών συστημάτων: Linux, Windows, OS X. Για όλες, προκειμένου να επιτευχθεί η σύνδεση πρέπει να είμαστε εκτός LAN του server.

Περιπτωση Linux. Αν στη διανομή είναι εγκατεστημένο το OpenVPN, δεν χρειάζεται τίποτε άλλο. Ανοίγουμε ένα τερματικό, μεταβαίνουμε στον κατάλογο όπου είναι αποθηκευμένα τα αρχεία client.conf, ca.crt, laptop.crt και laptop.key, κι από τον λογαριασμό του root ή μέσω του sudo δίνουμε κάτι σαν

```
$ sudo /usr/sbin/openvpn --config client.conf
```

Για να διακόψουμε τη σύνδεση προς τον OpenVPN server, στο παράθυρο/καρτέλα τερματικού όπου έχουμε τρέξει το openvpn πατάμε το συνδυασμό πλήκτρων [CTRL+C].

Περιπτωση Windows. Ένα δωρεάν εργαλείο που μπορούμε να χρησι-



Ένας δωρεάν, ωραιότατος (;) OpenVPN client για Windows είναι το "OpenVPN GUI for Windows». Δεν κάνουμε πλάκα, αυτό είναι το πλήρες όνομά του.

μοποιούμε για τις συνδέσεις μας, είναι το λεγόμενο OpenVPN GUI for Windows (http://openvpn.se). Κατά την εγκατάστασή του τα Windows θα εμφανίσουν προειδοποίηση για άγνωστο εκδότη οδηγών -- ή κάπως έτσι, τέλος πάντων. Πείτε τους να ησυχάσουν και ν' αφήσουν τη διαδικασία να προχωρήσει. Τα αρχεία ca.crt, laptop.crt, laptop.key και client.conf πρέπει να τα μεταφέρουμε στον φάκελο "c:\program files\openvpn\config" του προγράμματος. Το client.conf είναι απαραίτητο να το μετονομάσουμε σε client.ovpn (είναι καλή ιδέα να ρυθμίσετε τα Windows ώστε να δείχνουν τις επεκτάσεις *όλων* των αρχείων). Εκτελώντας το OpenVPN GUI for Windows, στο system tray εμφανίζεται σχετικό εικονίδιο. Με δεξί κλικ πάνω του --και με την προϋπόθεση ότι τα απαραίτητα αρχεία είναι στον φάκελο " c:\program files\openvpn\config"-- θα δούμε το όνομα της σύνδεσης VPN. Το επιλέγουμε και σε ελάχιστα δευτερόλεπτα έχουμε συνδεθεί.

Περίπτωση OS X. Ένας δωρεάν client για το OpenVPN είναι το tunnelblick (http://code.google.com/p/tunnelblick). Παλαιότερα είχαμε βρει κάποια ενοχλητικά bugs στο πρόγραμμα, οπότε έκτοτε χρησιμοποιούμε το επί πληρωμή Viscosity (www.thesparklabs.com/viscosity, κοστίζει 9 δολάρια). Στο πρόγραμμα αρκεί να δώσουμε το αρχείο client.conf (πρέπει να βρίσκεται μέσα στον φάκελο με τα ca.crt, laptop.crt και laptop.key) και το προγραμματάκι αμέσως θα προσθέσει ένα νέο entry στο μενού των διαθέσιμων συνδέσεων. Επιλέγοντάς το συνδεόμαστε απευθείας στον αντίστοιχο OpenVPN server.

Αναδρομολόγηση *όλου* του traffic μέσα από το VPN

Με τις συνδέσεις που πετυχαίνουμε εφαρμόζοντας τα προηγούμενα είναι σαν να υλοποιούμε ένα ασφαλές, κρυπτογραφημένο κανάλι επικοινωνίας μεταξύ client (π.χ., το laptop μας) κι απομακρυσμένου OpenVPN server. Κατ' αυτόν τον τρόπο ο client γίνεται μέλος του LAN στο οποίο βρίσκεται ο server, έτσι έχουμε πρόσβαση σε κοινόχρηστους φακέλους, εκτυπωτές και γενικά σε υπηρεσίες εντός LAN. Όλα αυτά είναι εφικτά χάρη σ' ένα ψευδοδικτυακό ενδιάμεσο (pseudo network interface) που στήνει το OpenVPN κι ονομάζεται tun device. Μπορούμε έτσι να φανταζόμαστε ότι υπάρχει ένα ασφαλές τούνελ που ξεκινά από τον client και τερματίζει στον server.

Όμως το traffic όλων των άλλων δημόσιων interfaces *δεν* διακινείται μέσω του ασφαλούς αυτού τούνελ. Για παράδειγμα, η ασύρματη κάρτα δικτύωσης με την οποία έχουμε συνδεθεί στο ανοικτό hotspot της πλατείας, αποτελεί ένα δημόσιο network interface κι όλο το HTTP traffic διακινείται μέσω αυτού -- όχι μέσω του ασφαλούς tun device. Ιδανικά, αυτό που θα θέλαμε είναι όλο --μα όλο-- το network traffic να διέρχεται μέσω του tun, μ' άλλα λόγια μέσω του ασφαλούς, κρυπτογραφημένου τούνελ μεταξύ laptop και OpenVPN server. Ευτυχώς κάτι τέτοιο επιτυγχάνεται σχετικά εύκολα. Το μόνο που χρειάζεται είναι μια-δυο προσθήκες στο αρχείο server.conf, η ενεργοποίηση του λεγόμενου IP forwarding στον πυρήνα του Linux, καθώς κι ο ορισμός μερικών κανόνων για το iptables, ώστε τα δικτυακά πακέτα να διακινούνται σωστά κι ανεμπόδιστα. Όλα τα προαναφερθέντα επιτυγχάνονται πολύ πιο εύκολα απ' όσο αρχικά φαντάζεται κανείς. Ας στραφούμε λοιπόν ξανά στον OpenVPN server, ώστε να δούμε όλες τις λεπτομέρειες.

Βήμα 1. Θέλουμε το box με το OpenVPN να εκτελεί *και* χρέη nameserver για τους clients. Ακριβώς γι' αυτό θα χρειαστούμε τον DNS

proxy ονόματι dnsmasq:

```
root@oneiricvpn:~# apt-get -y install dnsmasq
```

Αμέσως μετά την εγκατάστασή του οφείλουμε να βεβαιωθούμε ότι αφουγκράζεται για αιτήσεις πελατών από το loopback interface (127.0.0.1) καθώς κι από τη διεύθυνση 10.8.0.1, που εξ ορισμού είναι είναι εκείνη του tun gateway. Ανοίγουμε λοιπόν μ' έναν text editor το αρχείο /etc/dnsmasq.conf κι αλλάζουμε τη γραμμή

```
#listen-address=
```

σε

```
listen-address=127.0.0.1, 10.8.0.1
```

Επίσης, από τ' αριστερά της γραμμής

```
#bind-interfaces
```

βγάζουμε το σύμβολο #, επομένως η γραμμή γίνεται

```
bind-interfaces
```

Αποθηκεύουμε τις αλλαγές στο /etc/dnsmasq.conf κι επανεκκινούμε την υπηρεσία:

```
root@oneiricvpn:/etc/openvpn# service dnsmasq restart
 * Restarting DNS forwarder and DHCP server dnsmasq [ OK ]
```

Βήμα 2. Ανοίγουμε το αρχείο /etc/openvpn/server.conf μ' έναν text editor και στο τέλος του προσθέτουμε τις ακόλουθες δύο γραμμές:

```
push "redirect-gateway def1"
push "dhcp-option DNS 10.8.0.1"
```

Αποθηκεύουμε τις αλλαγές, εγκαταλείπουμε τον editor κι επανεκκινούμε τον OpenVPN server:

```
root@oneiricvpn:~# service openvpn restart
* Stopping virtual private network daemon(s)...
*   Stopping VPN 'server'                         [ OK ]
* Starting virtual private network daemon(s)...
*   Autostarting VPN 'server'                     [ OK ]
```

Βήμα 3. Για την ενεργοποίηση του IP forwarding αρκεί να πληκτρολογήσουμε

```
root@oneiricvpn:~# echo "1" > /proc/sys/net/ipv4/ip_forward
```

Φυσικά, το IP forwarding θέλουμε να ενεργοποιείται και κάθε φορά που φορτώνει το λειτουργικό. Ακριβώς γι' αυτό ανοίγουμε το αρχείο /etc/sysctl.conf κι από τ' αριστερά της γραμμής

```
#net.ipv4.ip_forward=1
```

βγάζουμε τον χαρακτήρα #:

```
net.ipv4.ip_forward=1
```

Για άλλη μια φορά, αποθηκεύουμε την αλλαγή κι εγκαταλείπουμε τον editor.

Βήμα 4. Έναν προς έναν, δίνουμε στο τερματικό τους ακόλουθους κανόνες για το iptables:

```
root@oneiricvpn:~# iptables -A FORWARD -m state --state
RELATED,ESTABLISHED -j ACCEPT

root@oneiricvpn:~# iptables -A FORWARD -s 10.8.0.0/24 -j ACCEPT

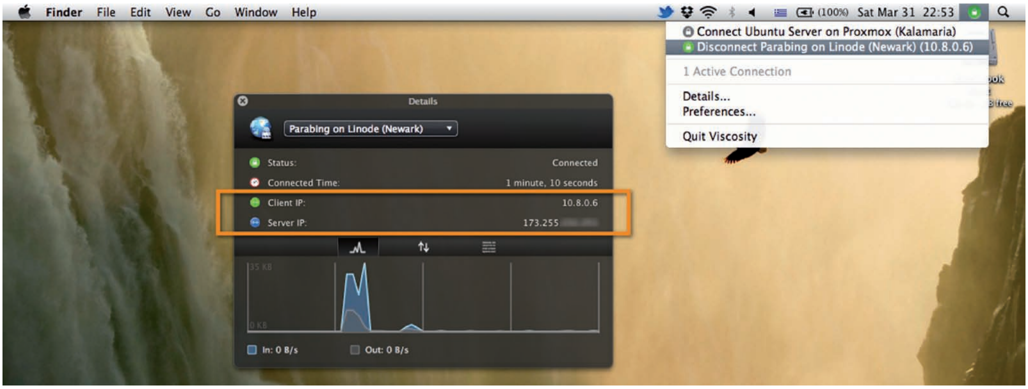
root@oneiricvpn:~# iptables -A FORWARD -j REJECT

root@oneiricvpn:~# iptables -t nat -A POSTROUTING -s 10.8.0.0/24
-o eth0 -j MASQUERADE
```

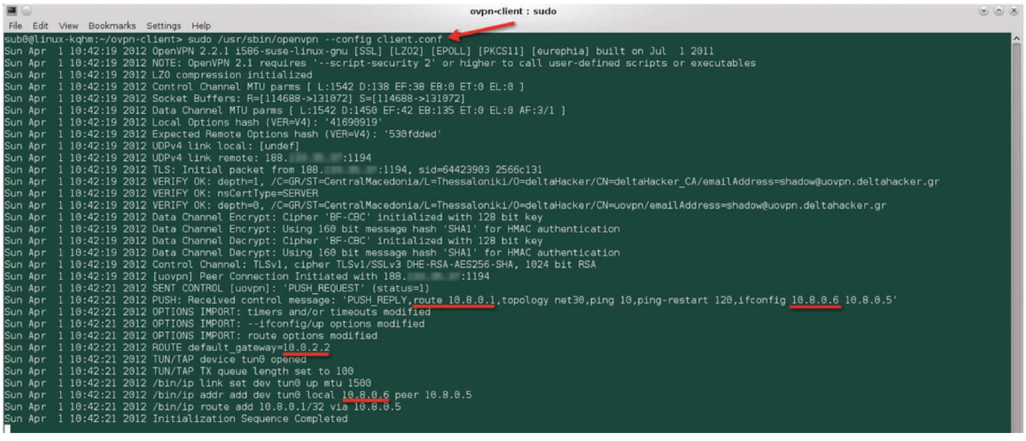
Βεβαίως, οι κανόνες θέλουμε να ενεργοποιούνται κάθε φορά που ξεκινά το λειτουργικό. Τους προσθέτουμε, λοιπόν, στο τέλος του αρχείου /etc/rc.local -- αλλά πριν την τελευταία γραμμή (exit 0). Δείτε, π.χ., με τι μοιάζει το δικό μας rc.local:

```
#!/bin/sh -e
#
# rc.local
#
# This script is executed at the end of each multiuser runlevel.
# Make sure that the script will "exit 0" on success or any other
# value on error.
#
# In order to enable or disable this script just change the
execution
# bits.
#
# By default this script does nothing.
iptables -A FORWARD -m state --state RELATED,ESTABLISHED -j
ACCEPT
iptables -A FORWARD -s 10.8.0.0/24 -j ACCEPT
iptables -A FORWARD -j REJECT
iptables -t nat -A POSTROUTING -s 10.8.0.0/24 -o eth0 -j
MASQUERADE
service dnsmasq restart
exit 0
```

Παρατηρήστε τη γραμμή



Ο OpenVPN client για το OS X που μας αρέσει και χρησιμοποιούμε, είναι το Viscosity. Μια δωρεάν εναλλακτική είναι το Tunnelblick.



Σύνδεση στον OpenVPN server από τη γραμμή εντολής του Linux. Πιο άμεσος και γρήγορος τρόπος σύνδεσης, απλά δεν υπάρχει :D

```
service dnsmasq restart
```

πριν από την τελευταία. Ο λόγος που την έχουμε βάλει είναι απλός: Κατά την εκκίνηση του Ubuntu Server το dnsmasq επιχειρεί να ξεκινήσει *πριν* ακόμα δημιουργηθεί το tun device, με αποτέλεσμα να αποτυγχάνει. Στο τέλος τις εκκίνησης του λειτουργικού, λοιπόν, τότε που διαβάζεται το αρχείο /etc/rc.local, είναι μια καλή ευκαιρία να επανεκκινούμε το dnsmasq!

Αυτό ήταν. Πλέον, κάθε φορά που χρησιμοποιούμε τον απομακρυσμένο OpenVPN server *όλη* η δικτυακή κίνηση διακινείται μέσω του tun device. Ένας τρόπος για να το επαληθεύσουμε είναι να κάνουμε δυο επισκέψεις σ' ένα site σαν το `http://whatismyipaddress.com`: Μία πριν συνδεθούμε στον OpenVPN server κι άλλη μία μετά, αφού συνδεθούμε.

> ***Σημείωση.*** *Επειδή στο απομακρυσμένο άκρο του τούνελ, εκεί δηλαδή που βρίσκεται ο OpenVPN server, κάποιος ενδέχεται να κάνει διάφορα περίεργα πράγματα όπως, π.χ., network sniffing, παρά την άριστη ασφάλεια που μας παρέχεται για το κομμάτι της διαδρομής μεταξύ client και server, όταν έχουμε να κάνουμε με ευαίσθητες υπηρεσίες οφείλουμε να χρησιμοποιούμε ασφαλή πρωτόκολλα (π.χ., HTTPS, SFTP κ.λπ.) ούτως ή άλλως.*

Η περίπτωση VPS

Το στήσιμο ενός VPN server εντός του οικιακού δικτύου προσφέρει άριστη ασφάλεια και ταυτόχρονα πλήρη πρόσβαση στο οικιακό LAN. Το μόνο αρνητικό αυτού του σεναρίου είναι ότι η ταχύτητα της σύνδεσής μας περιορίζεται από την ταχύτητα upstream του ADSL router (συνήθως 1Mbps στην καλύτερη περίπτωση).

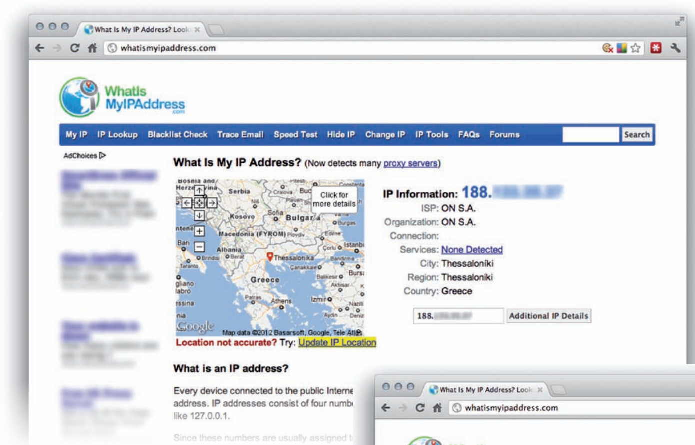
Αν μας ενδιαφέρει περισσότερο η ταχύτητα σύνδεσης από τη δυνατότητα πρόσβασης στο οικιακό LAN, τότε αξίζει να σκεφτούμε την περίπτωση στησίματος του OpenVPN πάνω σ' ένα VPS. Επιπρόσθετα, αναλόγως της γεωγραφικής τοποθεσίας του VPS, μέσω της σύνδεσης στον OpenVPN server είναι πιθανό ν' αποκτήσουμε πρόσβαση σε υπηρεσίες που προηγουμένως δεν μπορούσαμε να έχουμε -- κυρίως εξαιτίας γεωγραφικών περιορισμών που προκύπτουν λόγω ζητημάτων πνευματικής ιδιοκτησίας.

Όπως αναφέραμε και νωρίτερα, στο τεύχος 005 φιλοξενούμε άρθρο που

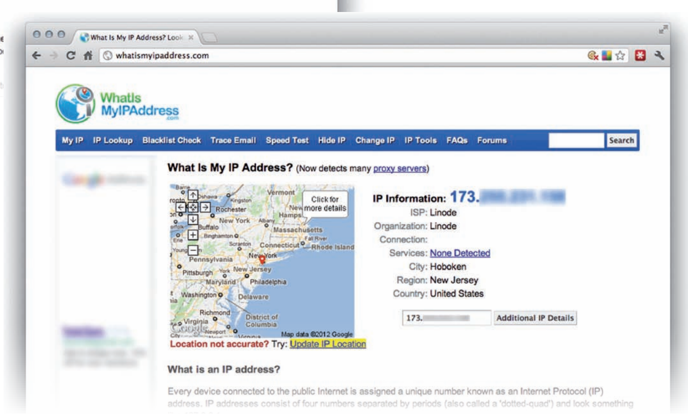
θα σας βοηθήσει να βρείτε και ν' αγοράσετε ένα φθηνό αλλά και ικανότατο VPS. Εμείς είχαμε ήδη ένα VPS σε data center κάπου στο Λονδίνο (σ' αυτό φιλοξενούμε το parabing.com). Η εταιρεία φιλοξενίας έχει data centers *και* στην Αμερική, με το γεωγραφικά εγγύτερο εξ αυτών να είναι στο New Jersey. Μεταξύ των παροχών που δίνει στους πελάτες της η εν λόγω εταιρεία, είναι και η αλλαγή data center. Ανοίγοντας σχετικό support ticket, λοιπόν, τους ζητήσαμε να μεταφέρουν το VPS μας από το Λονδίνο στο New Jersey -- και συγκεκριμένα στο Newark. Η μεταφορά ολοκληρώθηκε σε πέντε περίπου ώρες (από πριν μάς είχαν ενημερώσει ότι χρειάζονται περίπου 10 με 15 λεπτά ανά GB). Μετά από μερικές ρυθμίσεις στο host OS (Ubuntu Server 8.04 LTS) λόγω αλλαγής IP, στήσαμε το OpenVPN. Μπαίνοντας στο OpenVPN, ουσιαστικά αποκτήσαμε αμερικάνικο IP και συνεπώς είχαμε πρόσβαση σε όλες τις υπηρεσίες και site στα οποία έχουν πρόσβαση μόνο οι Αμερικάνοι. "Σπουδαίο κατόρθωμα", θα πουν κάποιοι εδώ -- και δεν θα 'χουν άδικο. Χάνουν όμως το νόημα όσοι λένε κάτι τέτοιο, ας μας επιτραπεί να παρατηρήσουμε: Η όλη ιδέα δεν είναι να ξημεροβραδιαζόμαστε στο Hulu ή στο Netflix, αλλά αν κι όταν θέλουμε να μπορούμε να το κάνουμε :) Ρίξτε αν θέλετε μια ματιά και στο σχετικό screencast που ετοιμάσαμε:

http://deltahacker.gr/?p=5115

Με γεια ο ολοκαίνουργιος OpenVPN server!



Σερφάρισμα χωρίς σύνδεση στο OpenVPN server. Παρατηρήστε διεύθυνση IP και γεωγραφική τοποθεσία.



Σερφάρισμα μέσω του OpenVPN server που λειτουργεί σε VPS μας, σε κάποιο datacenter στο New Jersey. Για άλλη μια φορά, παρατηρήστε IP και γεωγραφική τοποθεσία που φαίνεται να έχουμε...



Skill: Beginner

Tags: Deep web, Dark web, Tor, anonymity, bitcoins

Περιήγηση στο deep web

by GiannouG

Μπορούμε να φανταστούμε το Διαδίκτυο σαν ένα τεράστιο παγόβουνο. Η μύτη του αποτελείται από όλα τα site, τα οποία είναι εύκολα προσβάσιμα από τον καθένα μας και τα οποία επισκεπτόμαστε καθημερινά. Οι δικτυακοί τόποι όπως το facebook.com, το deltahacker.gr αλλά και οποιοσδήποτε άλλος περνάει από το μυαλό μας όταν ακούμε για το Internet, βρίσκονται στην κορυφή αυτού του φανταστικού παγόβουνου. Τα παγόβουνα, όμως, όπως όλοι γνωρίζουμε, κρύβουν το μεγαλύτερο μέρος τους *κάτω* από την επιφάνεια του νερού...

Τι κρύβεται κάτω από την επιφάνεια; Πώς μπορούμε να το δούμε και τι θα κερδίσουμε αν όντως το δούμε; Σ' αυτό το άρθρο θ' ασχοληθούμε και θα μιλήσουμε για την πιο σκοτεινή πλευρά του Internet. Είναι αρκετοί εκείνοι που υποστηρίζουν ότι το Internet έχει πολλά επίπεδα, μερικά από τα οποία δεν είναι καν προσβάσιμα με τους συμβατικούς τρόπους. Για να πούμε την αλήθεια, αυτό δεν ξέρουμε αν ισχύει ή όχι, ενώ οι σχετικές πληροφορίες είναι λιγοστές και συνήθως αλληλοαναιρούνται. Αρχικά, θα μιλήσουμε για τα δύο κύρια μέρη του Internet: το εμφανές και το όχι-και-τόσο εμφανές. Ακολούθως, θ' ασχοληθούμε εκτενώς με το λεγόμενο Darknet!

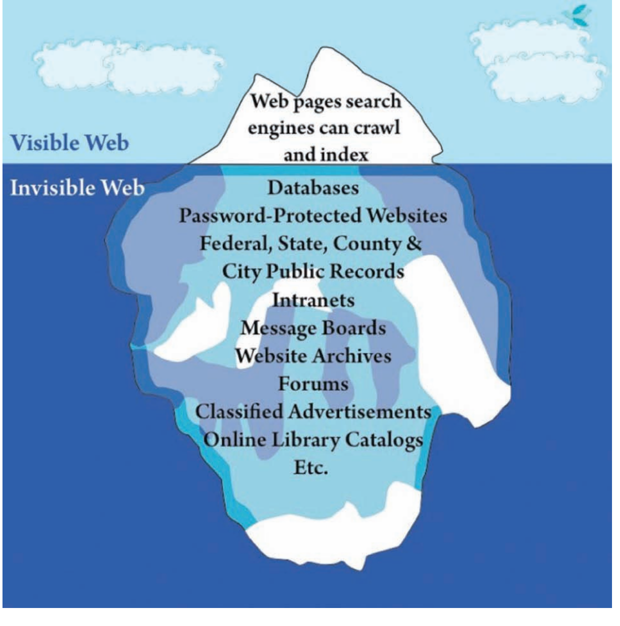
Common web, surface web, indexable web ή visible web

Ούτε οι μαφιόζοι δεν έχουν τόσα πολλά ονόματα! Πριν ξεκινήσουμε, να σας προειδοποιήσουμε ότι γίνεται χαμός με τους όρους, οι οποίοι δεν είναι και τόσο ξεκάθαροι αφού κάθε πηγή τους προσδιορίζει με διαφορετικό τρόπο. Ας επιστρέψουμε και πάλι στην εικόνα με ολόκληρο το παγόβουνα. Είπαμε ότι το εμφανές μέρος του αποτελείται απ' όλα τα γνωστά site. Όλοι αυτοί οι τόποι αποτελούν ένα μεγάλο μέρος του Διαδικτύου, όπως το ξέρουμε. Οι μηχανές αναζήτησης μπορούν να τους βρουν, να τους επεξεργαστούν και να περάσουν όλες τις πληροφορίες στα συστήματα τους. Μιλάμε για site, δηλαδή, τα οποία μπορεί να επισκεφτεί ο καθένας, βρίσκοντάς τα σε κάποια μηχανή αναζήτησης.

Invisible web, deepnet, undernet, deep web ή απλά hidden web

Εδώ τουλάχιστον έχει μαφιόζους ;) Πρόκειται για το μέρος του Διαδικτύου το οποίο *δεν* είναι εμφανές στις μηχανές αναζήτησης. Προσοχή! Αυτό δε σημαίνει ότι τα σχετικά site είναι απαραιτήτως κρυμμένα. Ο όρος "hidden web" αναφέρεται στα site που *αδυνατούν* να καταγράψουν οι μηχανές αναζήτησης. Μπορεί, για παράδειγμα, να πρόκειται για ιστοσελίδες οι οποίες αλλάζουν περιεχόμενο, ανάλογα με το ερώτημα το οποίο στέλνουμε στη βάση τους. Στο deep web συγκαταλέγονται πολλών ειδών site. Όπως είπαμε παραπάνω, εμείς θα ασχοληθούμε με τα site εκείνα τα οποία απαρτίζουν το λεγόμενο Darknet.

Κάπως έτσι μπορούμε να φανταζόμαστε το Internet. Η κορυφή αποτελείται απ' όλα τα γνωστά site, ενώ κάτω από την επιφάνεια... Ε, για το τι κρύβεται κάτω από την επιφάνεια διαβάστε το άρθρο.



Σύνδεση στο Darknet

Για να είμαστε ακριβείς, υπάρχουν αρκετά δίκτυα που χαρακτηρίζονται έτσι. Μερικά από αυτά είναι το I2P (www.i2p2.de), το Freenet (https://freenetproject.org) και φυσικά το Tor (www.torproject.org). Στο Darknet τα πράγματα σοβαρεύουν. Οι σελίδες δεν απαγορεύουν απλά την πρόσβαση στις μηχανές αναζήτησης, αλλά κρύβονται και πίσω από την ανωνυμία. Κάπως έτσι το πράγμα αρχίζει ν'

αποκτά πολύ, μα πάρα πολύ ενδιαφέρον! Στο Darknet μπορεί να βρει κανείς οποιαδήποτε πληροφορία ή/και αρχεία: Από πολύ ενδιαφέροντα βιβλία, απαγορευμένες ταινίες και μουσική, έως πληροφορίες τις οποίες καμία κυβέρνηση δεν θα ήθελε να διαρρεύσουν. Το μόνο σίγουρο είναι ότι αν χτυπήσει κάποιος φλέβα, θα περάσει πολλές ώρες διαβάσματος. Περισσότερα για όλα αυτά θα πούμε αργότερα. Για την ώρα, θα πούμε δυο λόγια για το δίκτυο Tor, καθώς είναι ιδιαίτερα εύχρηστο και μάλλον ιδανικό για τα πρώτα μας βήματα στο Darknet.

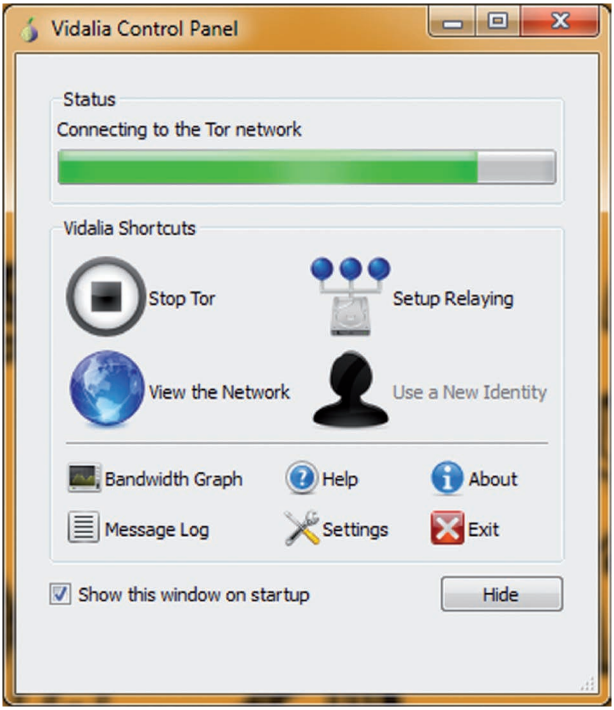
Το Tor εν συντομία

Κατά 99.987% όλοι όσοι διαβάζουν αυτήν την πρόταση έχουν ξανακούσει για το Tor, το δίκτυό του αλλά και το πώς λειτουργεί. Αν όχι, δεν πειράζει. Το Tor είναι ένα δίκτυο υπολογιστών (δουλεύει φυσικά εντός του Internet) που προσφέρει ανωνυμία στους χρήστες του. Όσοι υπολογιστές έχουν το πρόγραμμα του Tor εγκατεστημένο, μπορούν με την κατάλληλη επιλογή να μετατραπούν σε Tor relays (αναμεταδότες δεδομένων) ή σε Tor exits (έξοδοι προς το φανερό Internet). Όλα αυτά βέβαια δε λένε και πολλά σε όσους δεν έχουν ξανακούσει για το Tor. Ας δούμε λοιπόν τι συμβαίνει όταν είμαστε συνδεμένοι στο εν λόγω δίκτυο. Όταν προσπαθούμε να δούμε μια σελίδα του "κανονικού" Διαδικτύου[1] μέσα από το Tor, δημιουργείται δυναμικά μια τυχαία αλυσίδα από Tor relays. Το αίτημά μας ταξιδεύει κρυπτογραφημένο μέσα από αυτά τα relays και καταλήγει στο clearnet, περνώντας από κάποιο Tor exit. Σε κάθε νέο request το δίκτυο των Tor relays αλλάζει και φυσικά με εντελώς τυχαίο τρόπο. Έτσι, κανείς δεν μπορεί να μας εντοπίσει! Μέσα στο δίκτυο του Tor, το οποίο ονομάζεται και Onion, υπάρχουν κρυμμένες ορισμένες υπηρεσίες, οι οποίες διευθετούν όλα τα ζητήματα με τη διακίνηση αιτημάτων και φυσικά δεδομένων. Έτσι, όταν θέλουμε να επισκεφτούμε μια σελίδα η οποία βρίσκεται

Τι σχέση μπορεί να έχουν οι Anonymous με το darknet; Πριν από μερικούς μήνες είχαν γίνει επιθέσεις σε servers οι οποίοι διατηρούν παράνομο πορνογραφικό υλικό...



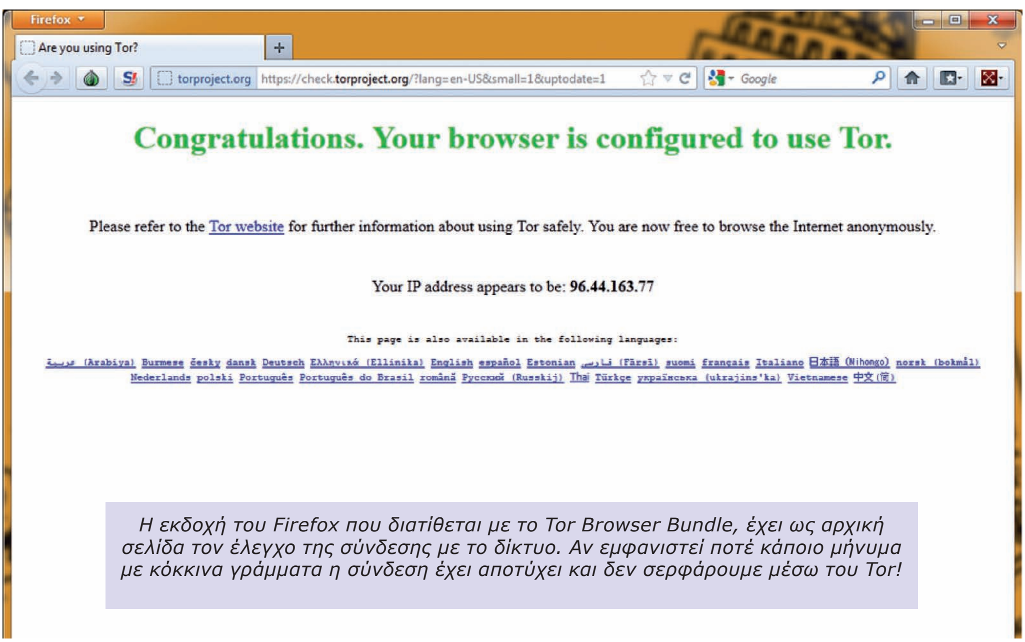
1. Από δω και πέρα, για να μιλάμε στη γλώσσα του deep web, το φανερό Internet θα το λέμε clearnet.



Το Vidalia (GUI για το Tor) προσπαθεί να συνδεθεί στο Onion. Προσοχή στο κουμπάκι «Use a New Identity”.

στο κρυμμένο δίκτυο του Tor, το αίτημά μας θα διακινηθεί μέσα από Tor relays αλλά δεν φτάσει ποτέ σε κάποιο Tor exit.

Καλά όλα αυτά, θα πείτε, αλλά πώς μπορούμε να τα γνωρίσουμε από πρώτο χέρι; Για αρχή, χρειαζόμαστε το Tor Browser Bundle. Πρόκειται για μια συλλογή προγραμμάτων, τα οποία διευκολύνουν τη ζωή μας στο Tor. Κανονικά, για τη σύνδεση στο Tor με τον browser του συστήματός μας απαιτείται η ρύθμιση διαφόρων “περιφερειακών” προγραμμάτων και δεν αρκεί μια απλή εγκατάσταση. Ωστόσο, το Tor Browser Bundle καταφέρνει αυτό που όλοι οι νεοφερμένοι ονειρεύονται: Με ένα download, μια αποσυμπίεση και μια εκτέλεση, εμφανίζεται μπροστά μας μια φορητή έκδοση Firefox που συνδέεται αυτόματα στο Tor! Αυτή η ειδική εκδοχή του Firefox είναι ρυθμισμένη ώστε να μην κρατάει καθόλου ιστορικό, cookies ή άλλα δεδομένα, ενώ ταυτόχρονα έχει εγκατεστημένα κάποια add-on για πρόσθετη ασφάλεια. Το Tor Browser Bundle μπορούμε να το κατεβάσουμε από



Η εκδοχή του Firefox που διατίθεται με το Tor Browser Bundle, έχει ως αρχική σελίδα τον έλεγχο της σύνδεσης με το δίκτυο. Αν εμφανιστεί ποτέ κάποιο μήνυμα με κόκκινα γράμματα η σύνδεση έχει αποτύχει και δεν σερφάρουμε μέσω του Tor!

την επίσημη σελίδα του Tor (www.torproject.org) και είναι διαθέσιμο για όλες τις πλατφόρμες. (Σ.τ.Ε. Περισσότερα για την εγκατάσταση και χρήση του Tor Bundle κάτω από Windows μπορείτε να διαβάσετε και στο http://parabing.com/?p=41)

Πρώτα βήματα στο deep web

Πριν ξεκινήσουμε τις περιπέτειες είναι απαραίτητη μια προειδοποίηση: Στο deep web υπάρχουν κάθε είδους πληροφορίες κι ένα σωρό παράνομα πράγματα! Μπορεί κανείς να συναντήσει από οδηγίες για την κατασκευή μολότοφ (όχι τις απλές συνταγές, φυσικά) μέχρι σχέδια πυραύλων, σχέδια εκκένωσης πόλεων ή και οτιδήποτε περνάει από το μυαλό του[2]. Επιπλέον, υπάρχει περίπτωση πολλές από αυτές τις σελίδες να περιέχουν ιδιαίτερα ακατάλληλο περιεχόμενο, τόσο νομικά, όσο και ηθικά. Κάθε κλικ πρέπει να γίνεται με μεγάλη προσοχή. Το Tor παρέχει αρκετά μεγάλη κάλυψη από πλευράς ανωνυμίας, αλλά είναι στο χέρι του χρήστη ποιες σελίδες θα επιλέξει να επισκεφθεί και τι θα κάνει εκεί. Ο αρθογράφος αλλά και το περιοδικό δεν ευθύνονται για όσα ενδέχεται να συναντήσετε εκεί μέσα. Το μόνο σίγουρο είναι πως οι περισσότερες σελίδες προκαλούν...

Το deep web είναι συνώνυμο του χάους, ειδικά για ένα νέο χρήστη. Δεν υπάρχει αρχή, ούτε και τέλος. Φανταστείτε το σαν έναν σπάγκο που έχει γίνει μπάλα και δεν ξεμπλέκεται με τίποτα. Παρόλα αυτά, υπάρχουν δύο τρείς "αρχικές" σελίδες, οι οποίες θα μπορούσαν να χρησιμοποιηθούν σαν αφετηρία για την περιπλάνηση στο deep web. Εμείς, σαν πρωτάρηδες σ' αυτόν τον κόσμο, θα επισκεφθούμε το "Welcome, We've been expecting you!" (http://p3lr4cdm3pv4plyj.onion). Περιέχει πληροφορίες για τους νέους χρήστες, τι πρέπει να προσέχει κάποιος αλλά και πώς μπορούμε να χρησιμοποιήσουμε το GnuPG για κρυπτογράφηση.

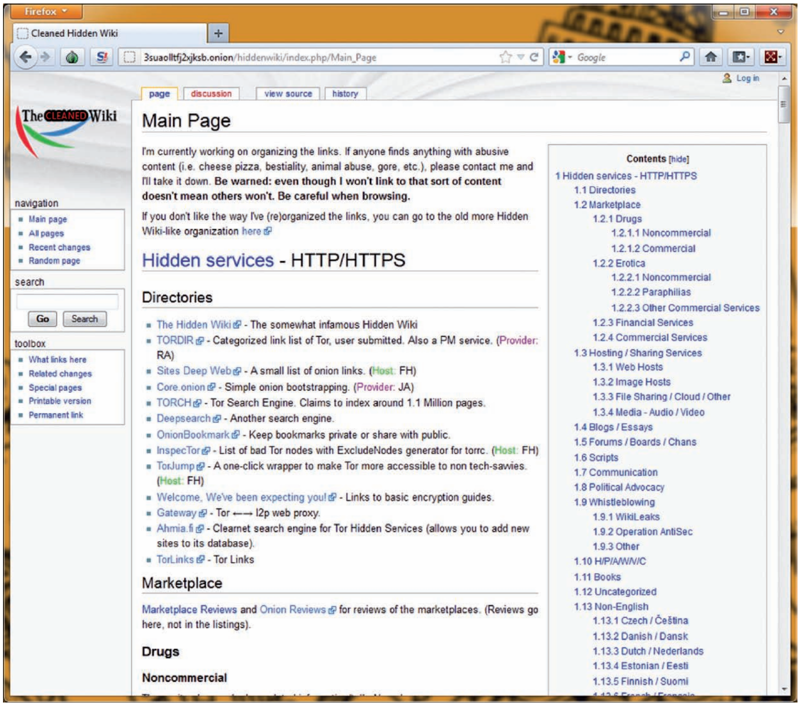
Ιστοσελίδες με ενδιαφέρον

Μια παρουσίαση του deep web θα ήταν μισερή, αν δεν περιελάμβανε μερικές αξιόλογες σελίδες με συμβουλευτικό και γενικότερα χρήσιμο περιεχόμενο. Οι περισσότερες από αυτές βρίσκονται σε διάφορους καταλόγους του Onion, μιας και αποτελούν κομβικά του σημεία. Εμείς κάναμε ένα πρώτο ξεκαθάρισμα, διαλέξαμε μερικές και μπορούμε να εγγυηθούμε πως είναι 100% "καθαρές" -- χωρίς αυτό να σημαίνει ότι πρέπει να μας εμπιστευτείτε στα τυφλά ;)

- http://eqt5g4fuenphqinx.onion (Core.onion): Όπως λέει και η αρχική του σελίδα, πρόκειται για entry-point στο δίκτυο του Tor από το 2007!
- http://kpvz7ki2v5agwt35.onion: Το πλέον γνωστό "The Hidden Wiki". Στις 100 φορές που θα προσπαθήσει κάποιος να μπει, τις 78.9 δεν θα τα καταφέρει[3]. Όπως τα περισσότερα site στο Onion, έτσι κι αυτό πέφτει πολύ συχνά, χωρίς προφανή λόγο[4]. Πρόκειται για τη μεγαλύτερη πηγή για νέα και φρέσκα site μέσα στο Onion. Προσοχή στο τι πατάτε!
- http://3suaolltfj2xjksb.onion/hiddenwiki: The CLEANED Wiki, για αυτούς που θέλουν να είναι σίγουροι πως ό,τι πατήσουν είναι καθαρό και δεν θα βρεθούν προ εκπλήξεων. Τίποτα παράνομο, απλή και καθαρή δόση deep web, για ατελείωτες ώρες αναζητήσεων.

2. Τα οποία κανείς δεν ξέρει αν είναι πραγματικά ή όχι. Μάλλον το δεύτερο, αλλά ας μη χαλάσουμε το κλίμα.
3. Την στιγμή που γράφεται το παρόν footnote, δεν δουλεύει!
4. Αν και το μυαλό μας πηγαίνει σε πολλά αίτια...

Το CLEANED Wiki θα μπορούσε άνετα να παίξει το ρόλο της αρχικής σελίδας του deep web. Αποτελεί την ιδανική αφετηρία, για κάθε νέα περιπέτεια σ' αυτό...



- http://dppmfxaacucguzpc.onion (TorDir): Μια κατηγοριοποιημένη λίστα με αρκετά link μέσα στο Onion. Όπως κάθε λίστα, σε visible αλλά και σε hidden web, είναι γεμάτη άκυρα link αλλά κρύβει και μερικά διαμάντια.

- http://xmh57jrzrnw6insl.onion (TORCH): Μια μηχανή αναζήτησης που υποστηρίζει ότι έχει αποτελέσματα για πάνω από 600.000 σελίδες του Onion. Η μοναδική σε λειτουργία μηχανή αναζήτησης -- αν και τη συγκεκριμένη στιγμή δε λειτουργεί ;)

- http://hpuuigeld2cz2fd3.onion (DeepSearch): Ακόμη μια μηχανή αναζήτησης, η οποία φορτώνει αρκετά γρήγορα. Άξια εναλλακτική του περίφημου TORCH, όταν αυτό δεν δουλεύει (και πιστέψτε μας, αυτό συμβαίνει συχνά).

- http://jhiwjjlqpyawmpjx.onion (TorMail): Ασφαλής αποστολή email τόσο μέσα, όσο και έξω από το δίκτυο του Tor. Οι διευθύνσεις είναι της μορφής user@tormail.net και απαιτείται σύνδεση μέσα από το δίκτυο του Tor για τη λειτουργία.

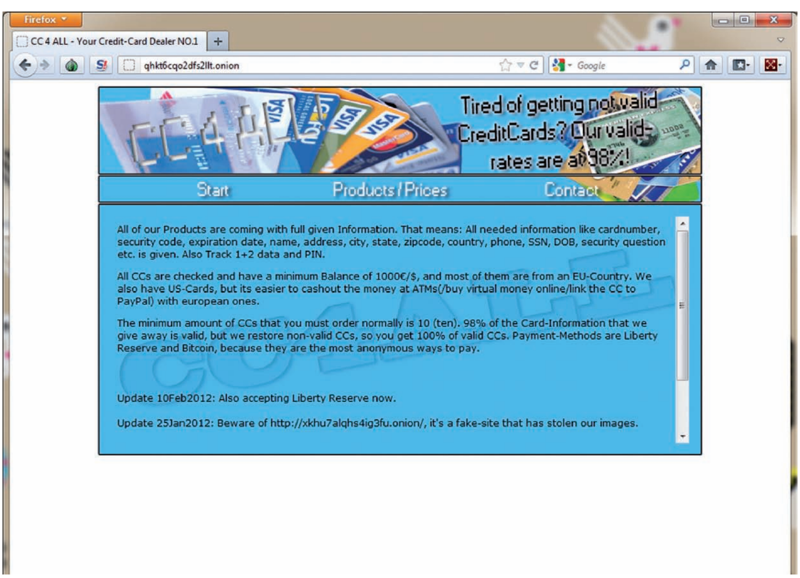
- http://4eiruntyxxbgfv7o.onion/pm (TorPM): Επικοινωνία μεταξύ χρηστών στο Onion. Σε κάθε χρήστη αντιστοιχεί ένα ID, το οποίο λειτουργεί σαν αναγνωριστικό για τη συνομιλία με άλλους. Το TorPM είναι ιδιαίτερα διαδεδομένο μεταξύ των συνειδητοποιημένων χρηστών του Onion..

- http://4eiruntyxxbgfv7o.onion (Circle): Υποστηρίζει αποθήκευση αρχείων, εικόνων αλλά και κειμένου, στο στιλ του pastebin!

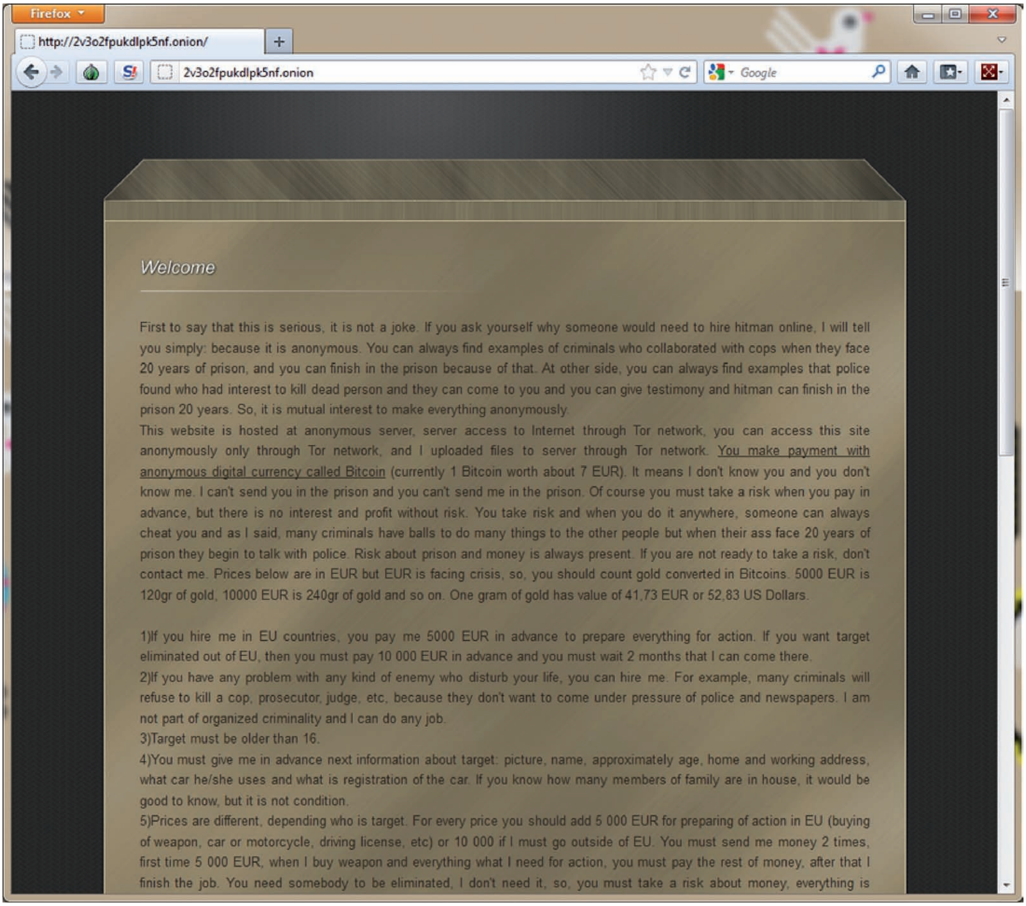
- http://zw3crggtadila2sg.onion/imageboard (TorChan): Ένα image-

board αποκλειστικά στο Onion, με ό,τι αυτό συνεπάγεται. Οι γνώστες του είδους θα το λατρέψουν ;)

- http://clsvtzwzdgzkjda7.onion (HackBB): Forum με θέματα γύρω από το hacking. Μιλάμε για πιο προχωρημένα θέματα, σε σχέση με αυτά που συναντά κανείς στο απλό Internet.
- http://ajqaivfxtqy3fdlr.onion/torbook (TorBook): Θα μπορούσε το Onion να μην έχει το δικό του social network; Ακούγεται αντιφατικό, αν το καλοσκεφτεί κανείς. Ωστόσο social network υπάρχει, το οποίο μάλιστα απαιτεί να είσαι ενεργός χρήστης για να μπορέσεις να γίνεις μέλος.
- http://lotjbov3gzzf23hc.onion (TorStatusNet): Το Twitter του Onion. Τι σας κάνει εντύπωση; Εδώ υπάρχει το ισοδύναμο του Facebook! Στα αθώα και σύντομα status updates κολλάτε;
- http://xqz3u5drneuzhaeo.onion (Freedom Hosting): Φιλοξενία ιστοσελίδων μέσα στο Tor. Η συγκεκριμένη υπηρεσία φιλοξενεί ένα μεγάλο μέρος των site του Onion, αλλά δεν είναι ανοιχτή σε όλους. Για να γίνει κανείς μέλος, χρειάζεται ειδική πρόσκληση. Αν βρείτε, μην ξεχάσετε να μας ειδοποιήσετε. Σοβαρά το λέμε.
- http://wf4df37hrebhwzts.onion (ToR Design Guidelines): Τι πρέπει να προσέχει κάποιος, ο οποίος θέλει να φτιάξει τη δική του σελίδα στο Onion; Η ασφάλεια είναι μεγάλη, αλλά το bandwidth περιορισμένο... Το ToR Design Guidelines καλύπτει όλα όσα πρέπει να γνωρίζει ο επίδοξος σχεδιαστής.
- http://utup22qsb6ebeejs.onion (My Hidden Blog): Το blog ενός άγνωστου τυπά! Δεν έχει καμία ιδιαίτερη σημασία, αλλά έχει σίγουρα ενδιαφέρον. Μοιράζεται κυρίως σκέψεις και αξίζει μια επίσκεψη.
- http://xqz3u5drneuzhaeo.onion/users/experiments (The Human Experiment): Πειράματα σε ανθρώπους. Αν και πρόκειται για πλάκα, έχει πολύ ενδιαφέρον. Από τις πιο γνωστές σελίδες στο Onion.



Στο σκοτεινό κόσμο του darknet τα πάντα είναι δυνατά. Μέχρι και κλεμμένες πιστωτικές κάρτες, μπορεί να αγοράσει κανείς από εδώ! Βέβαια οι πληρωμές δεν γίνονται με πιστωτική, αλλά με bitcoins.



Αντίζηλος; Κανένα πρόβλημα. Μέσα από το darknet είναι υπόθεση μερικών λεπτών να βρεθεί κάποιος εκτελεστής. Τώρα, το αν πρόκειται για αλήθεια ή πλάκα δεν θα το μάθουμε ποτέ, γιατί πολύ απλά δεν θα το δοκιμάσουμε ποτέ. Στο κάτω-κάτω δεν θέλουμε να ξέρουμε!

- `http://3terbsb5mmmdyhse.onion` (hashparty): Πώς θα μπορούσε να λείπει μια τέτοιου είδους σελίδα από το Onion; Στο hashparty βοηθάς και σε βοηθούν να σπάσεις --τι άλλο;-- hashes. Το σύστημα δουλεύει με credits: δίνεις για να σπάσεις κάποιο hash, παίρνεις όταν σπάσεις εσύ. Αρκετά ενδιαφέρον για τους λάτρεις του είδους.

Αγορές μέσα από το deep web

Ναι, αμέ. Γιατί όχι, δηλαδή; Αρκεί να προσπεράσετε τις αγγελίες για δολοφονίες, ναρκωτικά και κάθε είδους παρανομία. Ίσως κάποτε, για κάποιο λόγο, χρειαστεί να αγοράσετε κάτι, το οποίο διατίθεται μόνο στο deep web. Για παράδειγμα, ένα πρόγραμμα Trojan horse[5]! Αν τέλος πάντων αποφασίσετε να αγοράσετε κάτι, ο μοναδικός τρόπος για να διατηρήσετε την ανωνυμία σας, είναι να πληρώσετε με Bitcoin (`http://bitcoin.org`). Το Bitcoin αποτελεί ένα νέο ψηφιακό νόμισμα, σε δοκιμαστικό στάδιο. Για το Bitcoin δεν υπάρχει κεντρική διαχείριση και όλες οι συναλλαγές γίνονται peer to peer. Η λειτουργία του παρουσιάζει ιδιαίτερο ενδιαφέρον, αλλά ξεφεύγει από το θέμα μας. Πάντως μπορεί κανείς να πληρώσει και με το

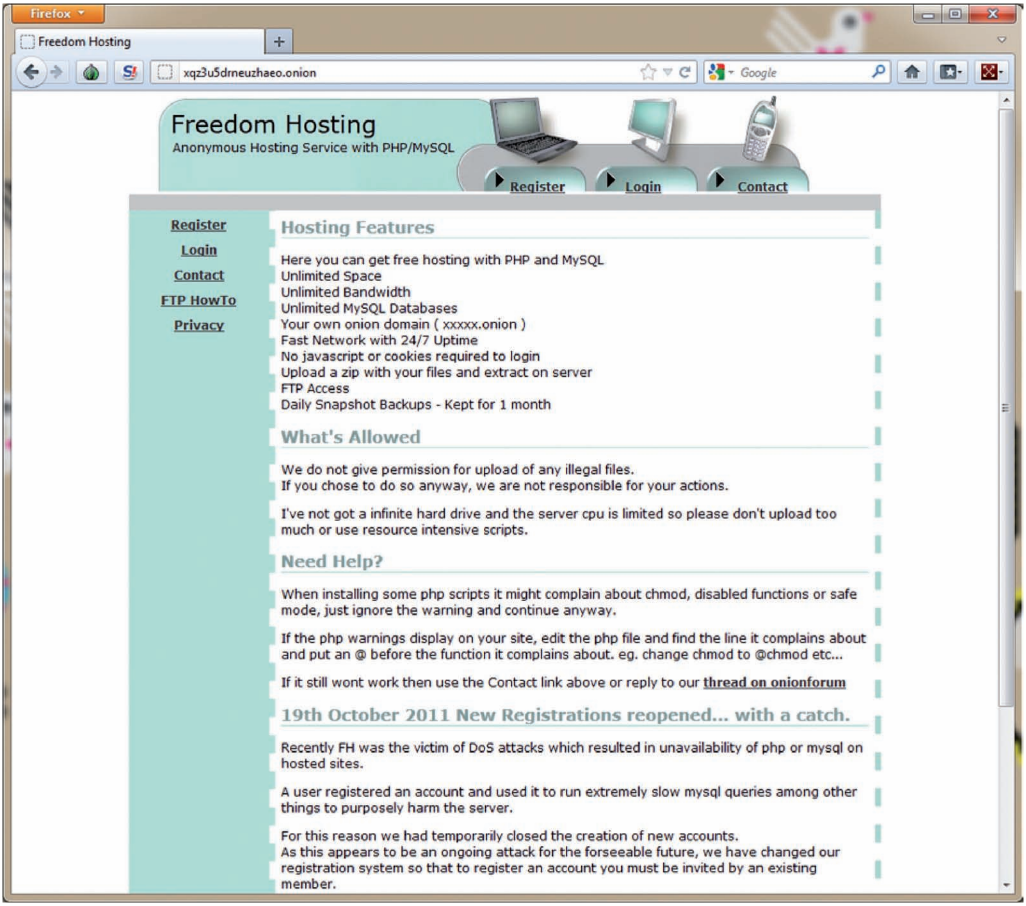
5. Καλύτερα όμως να μη σας μπαίνουν ιδέες. Αν θέλετε κάτι τέτοιο, φτιάξτε το μόνοι σας. Αν δεν μπορείτε, περιμένετε και θα το κάνουμε μαζί -- στο κοντινό μέλλον.

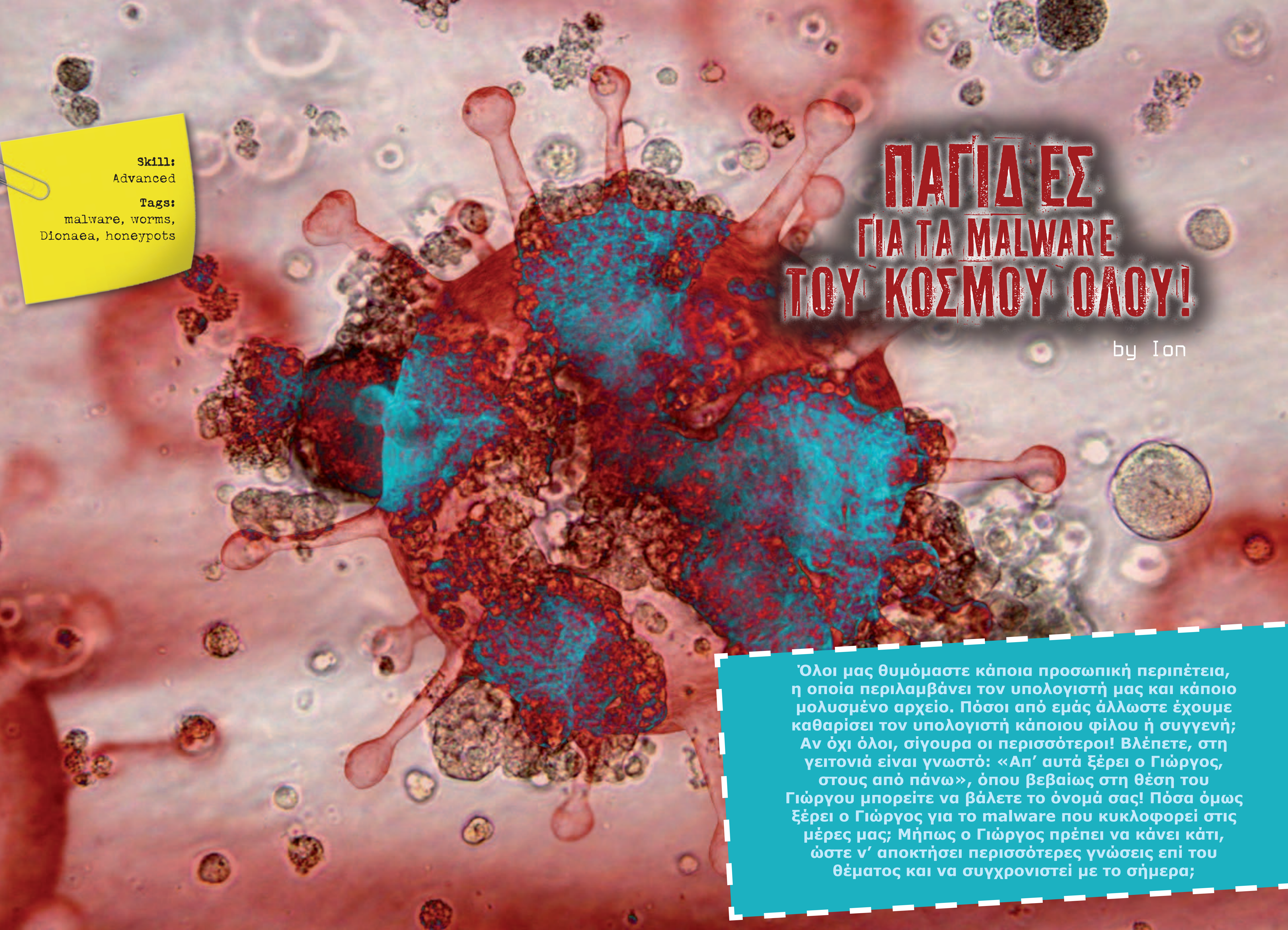
PayPal, αν και όπως καταλαβαίνετε κάτι τέτοιο δεν ενδείκνυται!

Καλές βόλτες!

Για τους περισσότερους, το deep web είναι κάτι πολύ πιο light από αυτό το οποίο φαντάζονταν. Οι περισσότεροι θα ήθελαν το deep web περισσότερο σκοτεινό. Εμείς, από τη μεριά μας, διατηρώντας την ψυχραιμία και τη λογική μας, το βρίσκουμε ενδιαφέρον και μάλλον χορταστικό. Σ' αυτό το δίκτυο, ο καθένας μπορεί να εκφράζεται ελεύθερα και χωρίς αναστολές. Έτσι, μπορούμε να βρούμε ανθρώπους με κοινά ενδιαφέροντα και να δώσουμε ή να πάρουμε άφθονες πληροφορίες, από εκείνες που δεν βρίσκει κανείς στο clean web! Βέβαια, όπως είπαμε επανειλημμένα, θα πρέπει να είμαστε προσεκτικοί σε κάθε μας βήμα. Επίσης, αν θελήσουμε να επιδοθούμε σε αγοραπωλησίες, θα ήταν φρόνιμο να γνωριστούμε και να εξοικειωθούμε με το Bitcoin. Η λειτουργία του παρουσιάζει ιδιαίτερο ενδιαφέρον και δεν αποκλείεται να μιλήσουμε σύντομα (και) για αυτό.

Το Freedom Hosting φιλοξενεί περίπου το 50% των σελίδων στο Tor. Τώρα καταλάβατε γιατί ήταν ο κεντρικός στόχος των Anonymous... ;)





Skill:
Advanced

Tags:
malware, worms,
Dionaea, honeypots

ΠΑΓΙΔΕΣ ΓΙΑ ΤΑ MALWARE ΤΟΥ ΚΟΣΜΟΥ ΟΛΟΥ!

by Ion

Όλοι μας θυμόμαστε κάποια προσωπική περιπέτεια, η οποία περιλαμβάνει τον υπολογιστή μας και κάποιο μολυσμένο αρχείο. Πόσοι από εμάς άλλωστε έχουμε καθαρίσει τον υπολογιστή κάποιου φίλου ή συγγενή; Αν όχι όλοι, σίγουρα οι περισσότεροι! Βλέπετε, στη γειτονιά είναι γνωστό: «Απ' αυτά ξέρει ο Γιώργος, στους από πάνω», όπου βεβαίως στη θέση του Γιώργου μπορείτε να βάλετε το όνομά σας! Πόσα όμως ξέρει ο Γιώργος για το malware που κυκλοφορεί στις μέρες μας; Μήπως ο Γιώργος πρέπει να κάνει κάτι, ώστε ν' αποκτήσει περισσότερες γνώσεις επί του θέματος και να συγχρονιστεί με το σήμερα;

Θα περίμενε κανείς ότι μετά από μερικά χρόνια εμπειρίας, είναι πια τόσες οι γνώσεις μας ώστε δεν κινδυνεύουμε από καμία Διαδικτυακή απειλή. Ωστόσο κάτι τέτοιο δεν ισχύει στην πραγματικότητα κι όποιος το ισχυριστεί καλά θα κάνει να 'ναι έξτρα προσεκτικός. Βλέπετε, σχεδόν καθημερινά εμφανίζεται νέο και πιο εξελιγμένο malware (malicious software). Αναρωτιέστε τι ακριβώς είναι αυτό; Πρόκειται για λογισμικό που κατασκευάζεται με δόλια κίνητρα, το οποίο διεισδύει σε ένα σύστημα χωρίς εξουσιοδότηση και εκτελεί κάποια ανεπιθύμητη εργασία. Το μεγάλο πρόβλημα με το malware είναι το εξής: Σε αντίθεση με έναν εισβολέα, ο οποίος θα έθετε σαν στόχο του δυο-τρία μηχανήματα, το malware στοχοποιεί ολόκληρα δίκτυα! Ένα καλά σχεδιασμένο malware αναζητά ακατάπαυστα υπολογιστές με συγκεκριμένες αδυναμίες και μπορεί να εξαπολύει αυτοματοποιημένες και εξειδικευμένες επιθέσεις σε εκατοντάδες μηχανήματα. Αναρωτιέστε ποιό είναι το κίνητρο πίσω από τη συγγραφή του malware; Κύριο μοχλός ανάπτυξης αποτελεί το κέρδος. Κάθε υπολογιστής που περνάει στον έλεγχο ενός ατόμου, μέσω της μόλυνσης από κακόβουλο λογισμικό, μπορεί να "πουληθεί" σε τρίτους ή και να αξιοποιηθεί από τον ίδιο το δημιουργό του malware, για μαζικές και φυσικά παράνομες ενέργειες (βλέπε (D)DoS attacks, αποστολή spam κ.ο.κ.). Υπάρχει ένα ενδιαφέρον βίντεο από το DEFCON 18, στο οποίο ένας πρώην προγραμματιστής malware εξιστορεί το πώς αναγκάστηκε να φτάσει εκεί και τι ακριβώς περιελάμβανε η "εργασία" του. Σας προτείνουμε να το παρακολουθήσετε: `http://bit.ly/spywaredevtalk`

Το malware χωρίζεται σε ορισμένες βασικές κατηγορίες: Στα προγράμματα που μολύνουν αυτόματα το ένα σύστημα μετά το άλλο (viruses και worms), σ' εκείνα που επιτρέπουν τον απομακρυσμένο έλεγχο των μολυσμένων συστημάτων (backdoors και trojan horses), καθώς και στα προγράμματα που στοχεύουν στο καθαρό διαφημιστικό κέρδος (spyware και adware). Προφανώς, υπάρχουν και όλοι οι συνδυασμοί αυτών των κατηγοριών. Γύρω από το θέμα θα μπορούσαν να γραφούν πολλές μελέτες και άρθρα -- αν δεν έχουν ήδη. Εμείς, στο παρόν άρθρο και σ' αυτό που ξεκινά από τη σελίδα 90, θα επικεντρωθούμε στα worms. Πρόκειται για το είδος malware που διακρίνεται από την έντονη προσπάθεια μετάδοσης του εαυτού του μέσα σ' ένα δίκτυο. Στην ουσία, τα worms αποτελούν ακούραστους εισβολείς, οι οποίοι αναζητούν συστήματα με συγκεκριμένες αδυναμίες. Όταν εντοπίζουν υποψήφια θύματα επιτίθενται κι εφόσον τα καταλάβουν συνεχίζουν από εκεί την αναζήτηση νέων θυμάτων. Όπως αντιλαμβάνεστε, τα worms προκαλούν συμφόρηση στα τοπικά δίκτυα και καταναλώνουν bandwidth, ενώ οι μολυσμένοι υπολογιστές ενδέχεται να συμμετάσχουν αργότερα και σε επιθέσεις DDoS (Distributed Denial of Service attacks).

Σε πολύ λίγο θα δείξουμε πώς στήνουμε ένα malware honeypot, με τη βοήθεια του οποίου θα προσπαθήσουμε να αιχμαλωτίσουμε διάφορα worms. Αυτό θα το κάνουμε για να συλλέξουμε όσο το δυνατόν περισσότερες πληροφορίες γύρω από τη δράση τους, ώστε ν' αποκτήσουμε καλή γνώση των μεθόδων επίθεσης. Με την ευκαιρία, θα πάρουμε και μια καλή γεύση από το περιβάλλον στο οποίο βγαίνουμε, κάθε φορά που συνδεόμαστε online. Απώτερος στόχος μας, όπως αντιλαμβάνεστε, είναι η απόκτηση γνώσεων, οι οποίες θα μας επιτρέψουν να αμυνόμαστε καλύτερα! Για την κατασκευή του honeypot θα μας βοηθήσει ένα ειδικό πακέτο λογισμικού, το οποίο κατασκευάζεται ακριβώς για το σκοπό που μας ενδιαφέρει. Πρόκειται για το λεγόμενο Dionaea (`http://dionaea.carnivore.it`). Αν ακούτε πρώτη φορά το όνομα, να σας πούμε ότι πρόκειται για

ένα σαρκοφάγο φυτό. Εδώ που τα λέμε, η δράση του honeypot που θα στήσουμε δεν θα απέχει και πολύ από εκείνη των σαρκοφάγων φυτών, τα οποία καταπίνουν τα ανυποψίαστα έντομα ;)

Dionaea malware honeypot

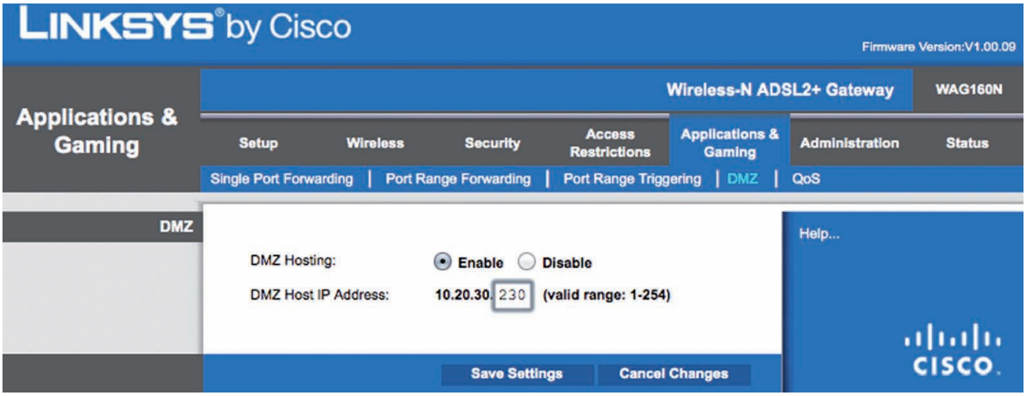
Το Dionaea αποτελεί ένα low interaction honeypot (βλ. και deltaHacker 006, σελ. 12). Όπως υποψιάζεστε, προσομοιώνει διάφορες ευπαθείς (vulnerable) υπηρεσίες τις οποίες στοχεύουν τα περισσότερα worm. Η ανάπτυξη του Dionaea βασίστηκε σ' ένα άλλο γνωστό malware catcher, με το όνομα Nepenthes. Αποστολή του Dionaea είναι ν' αποκτά αντίγραφα των ίδιων των malware. Έτσι, θα μπορούμε αργότερα να τα αναλύσουμε με διάφορες τεχνικές, να μαθαίνουμε πώς λειτουργούν και πώς μπορούμε να προστατευόμαστε από αυτά. Το Dionaea υλοποιεί διάφορα πρωτόκολλα και υπηρεσίες, στηρίζεται δε στη γλώσσα Python και σε μερικές πρόσθετες βιβλιοθήκες. Το βασικό πρωτόκολλο που ανοίγει προς τα έξω το Dionaea είναι το SMB (Server Message Block), επίσης γνωστό με την ονομασία CIFS (Common Internet File System). Πρόκειται για τα πλέον διαδεδομένα πρωτόκολλα σε file server οι οποίοι εξυπηρετούν μηχανήματα με Windows. Στο Linux, την αντίστοιχη υπηρεσία προσφέρει ο γνωστός SAMBA server. Το SMB τρέχει στη θύρα 445, έχει μια μακροσκελή ιστορία από exploits κι αποτελεί τον αγαπημένο στόχο των malware. Αυτό, όμως, θα το παρατηρήσετε και μόνοι σας σε λίγο...

Άλλα πρωτόκολλα που υλοποιεί το Dionaea είναι τα HTTP, FTP, TFTP, MSSQL, MySQL και SIP. Ανάμεσα σ' αυτά ξεχωριστό ενδιαφέρον παρουσιάζουν τα συστήματα βάσεων δεδομένων, καθώς και το πρωτόκολλο SIP, το οποίο χρησιμοποιείται σε υπηρεσίες VoIP. Αν σταθούμε τυχεροί, πέρα από τα malware που θα πιάσουμε θα δεχτούμε επιθέσεις και στις υπόλοιπες υπηρεσίες. Έτσι, θα μας δοθεί η ευκαιρία να μελετήσουμε πολύ περισσότερο υλικό.

Αν δεν στήνουμε το Dionaea honeypot σε VPS αλλά σε VM ή box εντός του τοπικού δικτύου, τότε καλό είναι να το βάλουμε στο DMZ (DeMilitarized Zone). Το Dionaea ανοίγει τόσες πολλές ευπαθείς υπηρεσίες προς τον έξω κόσμο, που τα port forwarding δεν έχουν νόημα.

Τρόπος λειτουργίας

Αφού ανοίξει τις απαραίτητες δικτυακές θύρες, το Dionaea περιμένει υπομονετικά για συνδέσεις από το Διαδίκτυο. Συνήθως αυτές δεν καθυστερούν καθόλου να έρθουν. Για την ακρίβεια, το πλήθος τους και η ταχύτητα με την οποία θα αρχίσουν να βομβαρδίζουν το Dionaea θα αποτελέσουν έκπληξη *και* για εσάς! Όπως καταλαβαίνετε, αυτές οι συνδέσεις δεν είναι καθόλου αθώες. Αντίθετα, επιχειρούν να εκμεταλλευτούν τις αδυναμίες των υπηρεσιών που προσομοιώνει το Dionaea. Τα διάφορα



malware θα ανοίξουν ένα κανάλι επικοινωνίας με το honeypot, θα ανταλλάξουν μερικά πακέτα και κατόπιν θα στείλουν τα αντίστοιχα payload. Ουσιαστικά, το payload αποτελεί τον κακόβουλο κώδικα, ο οποίος εκμεταλλεύεται την εκάστοτε αδυναμία. Φυσικά, τα payload δεν εκτελούνται κανονικά στο μηχάνημά μας, αλλά εντός του στεγανού περιβάλλοντος που έχει δημιουργήσει το Dionaea. Την ίδια στιγμή, το honeypot χρησιμοποιεί μια ειδική βιβλιοθήκη η οποία αναλαμβάνει να κάνει το λεγόμενο profiling, δηλαδή την καταγραφή όλων των κλήσεων συστήματος που πραγματοποιεί το εκάστοτε payload. Με αυτόν τον τρόπο αποκτούμε αρκετές πληροφορίες για τη δράση των malware, για τις ενδεχόμενες βλέψεις των δημιουργών τους κ.ο.κ.

Το payload που φτάνει κάθε φορά στο μηχάνημά μας, είναι διαφορετικό. Ωστόσο, όταν το Dionaea αντιληφθεί τις προθέσεις του εκάστοτε payload, κάνει τα πάντα ώστε ο επιτιθέμενος (το malware) να θεωρήσει ότι πέτυχε το στόχο του. Για παράδειγμα, ένα payload ενδέχεται να προσφέρει ένα shell στον επιτιθέμενο, με τις γνωστές μεθόδους bind και reverse connection του Metasploit Framework. Σ’ αυτήν την περίπτωση, το Dionaea παρέχει ένα ψεύτικο shell στον επιτιθέμενο και ανταποκρίνεται στις εντολές που δέχεται φυσιολογικά. Κάποια άλλα payload ενδέχεται να επιχειρούν την άμεση λήψη ενός αρχείου. Τότε, το Dionaea κατεβάζει κανονικά αυτό που του ζητήθηκε, αφού στις περισσότερες περιπτώσεις πρόκειται για το ίδιο το malware. Βέβαια, καλού-κακού, αμέσως μετά τη λήψη το Dionaea στέλνει ένα αντίγραφο του malware σε κάποιες online υπηρεσίες, οι οποίες πραγματοποιούν malware analysis (π.χ., στο Norman Sandbox της ομώνυμης Νορβηγικής εταιρίας ή στο Anubis, του Διεθνούς Εργαστηρίου Ασφαλών Συστημάτων ISECLAB). Σύντομα, από αυτές τις υπηρεσίες λαμβάνουμε μιαν απάντηση, με τις λεπτομέρειες γι’ αυτό ακριβώς που ψαρέψαμε.

Προαπαιτούμενα

Για να δημιουργήσουμε μια παγίδα για malware, θα χρειαστούμε αφενός το ειδικό λογισμικό που περιγράψαμε κι αφετέρου έναν υπολογιστή. Προτείνεται η χρήση ενός VPS (virtual private server), ο οποίος θα βρίσκεται μόνιμα online και θα καταγράφει τα malware που αλωνίζουν στον άναρχο χώρο του Διαδικτύου. Για το δικό μας παράδειγμα θα χρησιμοποιήσουμε ένα φτηνό, low-end VPS, το οποίο βρίσκεται σε κάποιο data center στην Αμερική. Το λειτουργικό σύστημα που επιλέξαμε είναι η διανομή Ubuntu Server 11.04 32bit. Εσείς βέβαια μπορείτε να χρησιμοποιήσετε οποιαδήποτε άλλη διανομή Linux, θα πρέπει όμως να είναι βασισμένη στο Debian: Έτσι θα αποφύγετε διάφορα προβλήματα συμβατότητας με το Dionaea. Εξάλλου, αν δεν έχετε VPS τότε μπορείτε να αγοράσετε ένα φθηνό ακολουθώντας και τις συμβουλές που έχουμε σε σχετικό άρθρο στο deltaHacker 005 (βλ. σελ. 44). Εναλλακτικά, μπορείτε να εργαστείσε σε εικονική μηχανή (virtual machine, VM) που θα δημιουργήσετε με το δωρεάν VirtualBox ή το εμπορικό VMware. Σ’ αυτή την περίπτωση, φροντίστε ώστε η κάρτα δικτύου του VM να βλέπει απευθείας τον router του τοπικού δικτύου (bridged networking), δώστε στο λειτουργικό static IP (βλ. deltaHacker 006, σελ. 81) και, φυσικά, ορίστε τους κατάλληλους κανόνες port forwarding στον router, ώστε οι υπηρεσίες που προσφέρει το honeypot να είναι προσβάσιμες *και* απο το Internet. Γι’ αυτό το τελευταίο, επειδή το Dionaea ανοίγει ένα σωρό υπηρεσίες στον έξω κόσμο, καλό θα ήταν να βάλετε το VM στο DMZ.

Σημείωση: *Σε περίπτωση που δεν εργαστείτε σε VPS αλλά σε VM ή*

*σε box *εντός* του τοπικού σας δικτύου, έχετε υπόψη ότι ο ISP σας για λόγους ασφαλείας ενδέχεται να μπλοκάρει ορισμένα ports. Πιθανό παράδειγμα αποτελεί το port 445, που όπως είπαμε χρησιμοποιείται από τα Windows (και τη SAMBA) για το μοίρασμα φακέλων. Σε μια τέτοια περίπτωση, malware που προσπαθούν να εκμεταλλευτούν αδυναμίες του πρωτοκόλου SMB/CIFS *δεν* θα καταφέρουν να φτάσουν στο Dionaea.*

Στη συνέχεια θα παρουσιάσουμε το στήσιμο του νέου μας honeypot. Για να μη δημιουργήσουμε προβλήματα εκεί που δεν υπάρχουν --αλλά και για λόγους απλότητας-- θα πραγματοποιήσουμε ολόκληρη τη διαδικασία από το λογαριασμό του root. Και μια τελευταία σημείωση: Για λόγους που θα γίνουν φανεροί αργότερα --συγκεκριμένα στο άρθρο που αρχίζει από τη σελίδα 90-- καλό είναι να έχετε εγκαταστήσει στο Ubuntu και να 'ναι ήδη ενεργοποιημένος κάποιος web server, όπως είναι ο nginx ή ο Apache.

Ρύθμιση συστήματος

Αφού συνδεθούμε για πρώτη φορά στο Ubuntu ως root (δώστε sudo su και πληκτρολογήστε το password του χρήστη σας), πρέπει να πραγματοποιήσουμε ορισμένες προπαρασκευαστικές διεργασίες. Ξεκινάμε, λοιπόν, αναβαθμίζοντας το λειτουργικό (αυτό θα ήταν φρόνιμο να το κάνετε πάντα και σε τακτά χρονικά διαστήματα):

```
root@dionaea:~# apt-get update && apt-get -y upgrade
```

Συνεχίζουμε θέτοντας τη σωστή ζώνη ώρας στο σύστημα:

```
root@dionaea:~# dpkg-reconfigure tzdata
Current default time zone: 'Europe/Athens'
Local time is now:      Wed Mar 28 20:37:48 EEST 2012.
Universal Time is now:  Wed Mar 28 17:37:48 UTC 2012.
```

Αυτό ήταν όλο! Η μηχανή μας είναι πλέον έτοιμη να δεχθεί το λογισμικό, το οποίο θα προσομοιώσει τη λειτουργία διαφόρων επιτηδευμένα ευάλωτων υπηρεσιών. Ξέρετε, από εκείνες που λιγουρεύονται τα malware ;)

Απαραίτητα πακέτα λογισμικού

Το Dionaea malware honeypot προϋποθέτει ορισμένα πακέτα για τη λειτουργία του. Η εγκατάσταση αυτών δεν είναι δύσκολη διαδικασία, αλλά είναι σχετικά χρονοβόρα. Καλό θα ήταν να την ακολουθήστε με προσοχή, ώστε να μην ξεχάσετε κάτι και υποχρεωθείτε να την επαναλάβετε εξαρχής. Ξεκινάμε εγκαθιστώντας κάποια απαραίτητα πακέτα:

```
root@dionaea:~# apt-get -y install libudns-dev libglib2.0-dev
libssl-dev libcurl4-openssl-dev libreadline-dev libsqlite3-dev
python-dev libtool automake autoconf build-essential subversion
git-core flex bison pkg-config
```

Δυστυχώς, εδώ σταματάει το αυτοματοποιημένο κομμάτι. Θα συνεχίσουμε εγκαθιστώντας τα υπόλοιπα πακέτα χειροκίνητα. Αυτό συμβαίνει είτε γιατί τα περισσότερα δεν υπάρχουν "πακεταρισμένα" είτε γιατί το Dionaea απαιτεί πολύ συγκεκριμένες εκδόσεις τους. Πάντως δεν υπάρχει κανένας λόγος ανησυχίας. Όπως λέει και το γνωστό ρητό: Real men do it command-line! Για να μην παρεκκλίνουμε από τον επίσημο οδηγό εγκατάστασης, όλα τα αρχεία, οι βιβλιοθήκες, καθώς και το ίδιο το Dionaea, θα εγκατασταθούν κάτω από τη διαδρομή /opt/dionaea/. Για αρχή, λοιπόν, θα πρέπει να δημιουργήσουμε αυτόν τον κατάλογο:

```
root@dionaea:~# mkdir /opt/dionaea
root@dionaea:~# cd /opt/dionaea
```

Προχωρούμε σε εγκατάσταση της βιβλιοθήκης liblcfg, η οποία χρησιμοποιείται από το Dionaea για το αρχείο ρυθμίσεων (configuration file):

```
root@dionaea:/opt/dionaea# git clone git://git.carnivore.it/
liblcfg.git liblcfg
root@dionaea:/opt/dionaea# cd liblcfg/code/
root@dionaea:/opt/dionaea/liblcfg/code# autoreconf -vi
root@dionaea:/opt/dionaea/liblcfg/code# ./configure --prefix=/opt/
dionaea
root@dionaea:/opt/dionaea/liblcfg/code# make install
root@dionaea:/opt/dionaea/liblcfg/code# cd ../../
```

Συνεχίζουμε εγκαθιστώντας τη βιβλιοθήκη libemu, η οποία προσομοιώνει τη λειτουργία ενός βασικού συστήματος x86, αναγνωρίζει τα payloads και, αν χρειαστεί, μπορεί και να τα εκτελέσει κιόλας! Μ' αυτόν τον τρόπο το Dionaea συλλέγει τον κακόβουλο κώδικα που στέλνουν τα malware, στην προσπάθειά τους να εκμεταλλευτούν τις ευάλωτες υπηρεσίες του honeypot.

```
root@dionaea:/opt/dionaea# git clone git://git.carnivore.it/
libemu.git libemu
root@dionaea:/opt/dionaea# cd libemu
root@dionaea:/opt/dionaea/libemu# autoreconf -vi
root@dionaea:/opt/dionaea/libemu# ./configure --prefix=/opt/
dionaea
root@dionaea:/opt/dionaea/libemu# make install
root@dionaea:/opt/dionaea/libemu# cd ..
```

Επόμενο βήμα είναι η εγκατάσταση της βιβλιοθήκης libnl, η οποία επιτρέπει την επικοινωνία μεταξύ διαφορετικών διαδικασιών του user level με διαδικασίες του kernel level, μέσω Netlink sockets. Προφανώς, ένας τέτοιος μηχανισμός απαιτείται για τη λειτουργία του Dionaea:

```
root@dionaea:/opt/dionaea# git clone git://git.infradead.org/
users/tgr/libnl.git
root@dionaea:/opt/dionaea# cd libnl
root@dionaea:/opt/dionaea/libnl# autoreconf -vi
root@dionaea:/opt/dionaea/libnl# export LDFLAGS=-Wl,-rpath,/opt/
dionaea/lib
root@dionaea:/opt/dionaea/libnl# ./configure --prefix=/opt/dionaea
root@dionaea:/opt/dionaea/libnl# make
root@dionaea:/opt/dionaea/libnl# make install
root@dionaea:/opt/dionaea/libnl# cd ..
```

Συνεχίζουμε εγκαθιστώντας τη βιβλιοθήκη libev. Αυτή χρησιμοποιείται από διάφορες διαδικτυακές εφαρμογές για την αποδοτικότερη λειτουργία τους, όταν δέχονται πολλές συνδέσεις ταυτόχρονα. Όπως καταλαβαίνετε, τα malware προκαλούν μεγάλο χαμό Εκεί Έξω (TM)!

```
root@dionaea:/opt/dionaea# wget http://dist.schmorp.de/libev/
Attic/libev-4.04.tar.gz
root@dionaea:/opt/dionaea# tar xfz libev-4.04.tar.gz
root@dionaea:/opt/dionaea# cd libev-4.04
```

```
root@dionaea:/opt/dionaea/libev-4.04# ./configure --prefix=/opt/dionaea
root@dionaea:/opt/dionaea/libev-4.04# make install
root@dionaea:/opt/dionaea/libev-4.04# cd ..
```

Το Dionaea απαιτεί την έκδοση 3.2 της Python η οποία, ωστόσο, δεν είναι εγκατεστημένη σε καμία σχεδόν διανομή Linux. Αυτό αποτελεί "σχεδιαστική" επιλογή των distributors, καθώς οι οικογένειες 2.x και 3.x της Python παρουσιάζουν μεγάλες διαφορές μεταξύ τους. Για τη χειροκίνητη εγκατάσταση της έκδοσης 3.2, αρκεί να δώσουμε τα ακόλουθα:

```
root@dionaea:/opt/dionaea# wget http://www.python.org/ftp/python/3.2.2/Python-3.2.2.tgz
root@dionaea:/opt/dionaea# tar xfz Python-3.2.2.tgz
root@dionaea:/opt/dionaea# cd Python-3.2.2/
root@dionaea:/opt/dionaea/Python-3.2.2# ./configure --enable-shared --prefix=/opt/dionaea --with-computed-gotos --enable-ipv6 LDFLAGS="-Wl,-rpath=/opt/dionaea/lib/ -L/usr/lib/x86_64-linux-gnu/"
root@dionaea:/opt/dionaea/Python-3.2.2# make
root@dionaea:/opt/dionaea/Python-3.2.2# make install
root@dionaea:/opt/dionaea/Python-3.2.2# cd ..
```

Αυτό ήταν! Πλέον, θα μπορείτε να λέτε ότι μεταγλωττίσατε την Python μόνοι σας ;) Επιπρόσθετα, θα χρειαστούμε την Cython. Πρόκειται για μια γλώσσα η οποία αποτελεί υπερσύνολο της Python και επιτρέπει τη συγγραφή επεκτάσεων σε C/C++.

```
root@dionaea:/opt/dionaea# wget http://cython.org/release/Cython-0.15.tar.gz
root@dionaea:/opt/dionaea# tar xfz Cython-0.15.tar.gz
root@dionaea:/opt/dionaea# cd Cython-0.15
root@dionaea:/opt/dionaea/Cython-0.15# /opt/dionaea/bin/python3 setup.py install
root@dionaea:/opt/dionaea/Cython-0.15# cd ..
```

Τελευταίο βήμα αποτελεί η εγκατάσταση της έκδοσης 1.1.1 της γνωστής βιβλιοθήκης libpcap. Με αυτή τη βιβλιοθήκη το Dionaea θα καταγράφει τη



Προσαρμογή του αρχείου ρυθμίσεων του Dionaea με τη βοήθεια του nano. Σ' αυτόν τον text editor, για να μεταβούμε σε συγκεκριμένη γραμμή πατάμε το συνδυασμό πλήκτρων [CTRL+_], στην προτροπή που εμφανίζεται κάτω δίνουμε τον αριθμό γραμμής και πατάμε [Enter].

δικτυακή κίνηση του συστήματος:

```
root@dionaea:/opt/dionaea# wget http://www.tcpdump.org/release/libpcap-1.1.1.tar.gz
root@dionaea:/opt/dionaea# tar xfz libpcap-1.1.1.tar.gz
root@dionaea:/opt/dionaea# cd libpcap-1.1.1
root@dionaea:/opt/dionaea/libpcap-1.1.1# ./configure --prefix=/opt/dionaea
root@dionaea:/opt/dionaea/libpcap-1.1.1# make
root@dionaea:/opt/dionaea/libpcap-1.1.1# make install
root@dionaea:/opt/dionaea/libpcap-1.1.1# cd ..
```

Εγκατάσταση του Dionaea malware honeypot

Μετά από την εγκατάσταση των προαπαιτούμενων πακέτων, μπορούμε επιτέλους να εγκαταστήσουμε το ίδιο το Dionaea honeypot. Βέβαια, πριν από την εγκατάσταση θα χρειαστεί μια μεταγλώττιση. Οπότε, καλά θα κάνετε να φανείτε *και πάλι* ιδιαίτερα προσεκτικοί. Αρχικά, θα πρέπει να κατεβάσουμε τον κώδικα από το σχετικό repository:

```
root@dionaea:~# cd /opt/dionaea
root@dionaea:/opt/dionaea# git clone git://git.carnivore.it/dionaea.git dionaea
root@dionaea:/opt/dionaea# cd dionaea
root@dionaea:/opt/dionaea/dionaea# autoreconf -vi
```

Στο επόμενο βήμα πρέπει να προσέξουμε τη χρήση των backslashes ("\"). Μετά από το καθένα πατάμε ένα [Enter] και συνεχίζουμε να γράφουμε από κάτω, κανονικά:

```
root@dionaea:/opt/dionaea/dionaea# ./configure --with-lcfg-include=/opt/dionaea/include/ \
--with-lcfg-lib=/opt/dionaea/lib \
--with-python=/opt/dionaea/bin/python3.2 \
--with-cython-dir=/opt/dionaea/bin \
--with-emu-include=/opt/dionaea/include \
--with-emu-lib=/opt/dionaea/lib \
--with-gc-include=/usr/include/gc \
--with-ev-include=/opt/dionaea/include \
--with-ev-lib=/opt/dionaea/lib \
--with-nl-include=/opt/dionaea/include \
--with-nl-lib=/opt/dionaea/lib \
--with-curl-config=/usr/bin \
--with-pcap-include=/opt/dionaea/include \
--with-pcap-lib=/opt/dionaea/lib
root@dionaea:/opt/dionaea/dionaea# make
root@dionaea:/opt/dionaea/dionaea# make install
root@dionaea:/opt/dionaea/dionaea# cd
```

Αυτό ήταν, το Dionaea βρίσκεται πλέον στον υπολογιστή μας! Ας δούμε πού βρίσκονται ορισμένα σημαντικά αρχεία και φάκελοι του νέου μας honeypot.

- **/opt/dionaea/bin.** Εδώ βρίσκεται το εκτελέσιμο αρχείο του

Dionaea, καθώς και άλλα χρήσιμα scripts τα οποία θα δούμε στη συνέχεια (readlogsqltree, gnuplotsql κ.λπ.).

- ***/opt/dionaea/etc/dionaea.*** Εδώ βρίσκεται το αρχείο ρυθμίσεων του Dionaea (ονομάζεται dionaea.conf), το οποίο θα πρέπει να επεξεργαστούμε αρχικά, για τη σωστή λειτουργία του malware honeypot.
- ***/opt/dionaea/var/log.*** Η θέση αποθήκευσης των αρχείων καταγραφής του Dionaea.
- ***/opt/dionaea/var/dionaea.*** Σε αυτόν τον κατάλογο θα βρούμε τη βάση δεδομένων SQLite (ονομάζεται logsql.sqlite), στην οποία καταγράφονται όλα τα στοιχεία από τη λειτουργία του honeypot.
- ***/opt/dionaea/var/dionaea/binaries.*** Χώρος αποθήκευσης όλων των malware που το σύστημά μας λαμβάνει από άλλους, μολυσμένους υπολογιστές.
- ***/opt/dionaea/var/dionaea/bistreams.*** Εδώ βρίσκονται τα bidirectional streams. Τα ειδικά αρχεία, δηλαδή, με τη ροή δεδομένων από και προς το honeypot. Υπάρχει ένα τέτοιο αρχείο για κάθε σύνδεση. Μάλιστα, με τη χρήση ενός τέτοιου αρχείου και το εργαλείο tcpreplay, μπορούμε να επαναλάβουμε την εκάστοτε επίθεση και να την παρακολουθήσουμε βήμα προς βήμα!

Πριν προχωρήσουμε στην επεξεργασία του αρχείου ρυθμίσεων του Dionaea, πρέπει να βάλουμε μια ακόμα πινελιά στο νέο μας honeypot. Συγκεκριμένα, θα εγκαταστήσουμε το p0f. Πρόκειται για ένα πρόγραμμα ικανό ν' αναγνωρίζει το λειτουργικό σύστημα που τρέχουν όσοι υπολογιστές συνδέονται στον δικό μας.

```
root@dionaea:~# apt-get install p0f
```

Περιττό να πούμε ότι το Dionaea συνεργάζεται με το p0f. Έτσι, για κάθε υπολογιστή από τον οποίο δεχόμαστε συνδέσεις και τελικώς "μολυνόμαστε" από κάποιο/α malware, το Dionaea καταγράφει και το λειτουργικό του!

Ρύθμιση του Dionaea

Για να επεξεργαστούμε το αρχείο ρυθμίσεων του Dionaea, πρέπει πρώτα να πραγματοποιήσουμε δύο απλές αλλά σημαντικές ενέργειες.

- Καταρχάς, πρέπει να εκτελέσουμε το προγραμματάκι ifconfig από την κονσόλα και να σημειώσουμε το όνομα του network interface που χρησιμοποιείται το Ubuntu για την επικοινωνία με το Internet. Συνήθως φαίνεται από το πεδίο inet addr, στο οποίο θα δείτε τη δημόσια (public) IP του υπολογιστή. Εδώ βέβαια χρειάζεται λίγη προσοχή. Μερικές φορές είναι άλλη η εικονική κάρτα με τη δημόσια IP και άλλη η κάρτα που χρησιμοποιείται τελικά (αυτό συνέβη με το δικό μας VPS). Σε κάθε περίπτωση, για να βεβαιωθείτε μπορείτε να κοιτάξετε στα πεδία RX bytes και TX bytes. Η σωστή κάρτα δικτύου, εκείνη που συνδέεται στο Διαδίκτυο, θα πρέπει να εμφανίζει σχετικά μεγάλους αριθμούς σ' αυτά τα πεδία.
- Κατόπιν, πρέπει να επισκεφτούμε την υπηρεσία VirusTotal, στο `www.virustotal.com`. Πρόκειται για μια υπηρεσία δωρεάν αναγνώρισης ιών και σκουληκιών, η οποία επιτρέπει την αποστολή των υπό εξέταση αρχείων. Όπως υποψιάζεστε, αυτή την υπηρεσία θα τη χρησιμοποιεί αυτόματα *και* το Dionaea. Για να γίνει όμως αυτό, οφείλουμε πρώτα να φτιάξουμε (αν δεν έχουμε ήδη) έναν λογαρια-

σμό στο VirusTotal και κατόπιν να σημειώσουμε το μοναδικό API key, το οποίο δημιουργήθηκε για το λογαριασμό μας. Αυτό το κλειδί εμφανίζεται στο profile μας στο VirusTotal, σ' ένα γκρίζο πλαίσιο. Το αντιγράφουμε και το κρατάμε κάπου, γιατί αργότερα θα χρειαστεί να το δώσουμε στο Dionaea.

Πλέον, είμαστε έτοιμοι να προχωρήσουμε στη ρύθμιση του honeypot. Το αρχείο ρυθμίσεων είναι αρκετά μεγάλο και σχετικά περίπλοκο. Επομένως, για μία ακόμα φορά, θα επιστήσουμε την προσοχή σας. Αρχικά, ανοίγουμε το αρχείο ρυθμίσεων με κάποιον plain-text editor, όπως είναι το nano:

```
root@dionaea:~# cd /opt/dionaea/etc/dionaea/
```

```
root@dionaea:/opt/dionaea/etc/dionaea# vi dionaea.conf
```

Μέσα από το nano, για να μεταβούμε σε συγκεκριμένη γραμμή πατάμε το συνδυασμό πλήκτρων [CTRL+_], δίνουμε τον αριθμό γραμμής και πατάμε το [Enter]. Μπορούμε λοιπόν να πραγματοποιήσουμε άνετα τις απαραίτητες παρεμβάσεις:

- Στη γραμμή 5 τροποποιούμε την τιμή levels, ώστε το Dionaea να μην καταγράφει το debug information:

  ```
  levels = "all,-debug"
  ```

- Στη γραμμή 111, αντικαθιστούμε την εμφανιζόμενη διεύθυνση email με τη δική μας. Αυτό είναι απαραίτητο, ώστε να λαμβάνουμε τα αποτελέσματα από τις υπηρεσίες malware analysis, τις οποίες χρησιμοποιεί το Dionaea.
- Στη γραμμή 166 αλλάζουμε το mode σε:

  ```
  mode = "manual"
  ```

- Στη γραμμή 167, φροντίζουμε ώστε να αναγράφεται το σωστό network interface:

  ```
  addrs = { eth0 = ["0.0.0.0"] }
  ```

- Πιθανώς να είναι eth0, γενικά όμως πρέπει να βάλουμε το network interface που χρησιμοποιεί ο υπολογιστής μας για τη σύνδεση στο Διαδίκτυο.
- Στη γραμμή 317 κάνουμε εισαγωγή του κλειδιού API, από την υπηρεσία VirusTotal:

  ```
  apikey = "εδώ-μπαίνει-το-API-Key-123..."
  ```

- Τέλος, προσέχουμε να μην υπάρχει διπλό slash (δηλαδή comment) στην αρχή της γραμμής 456. Με αυτόν τον τρόπο επιβεβαιώνουμε ότι θα γίνεται πράγματι η καταγραφή στη βάση δεδομένων SQLite. Επιπρόσθετα, πρέπει να αφαιρέσουμε το διπλό slash από την αρχή των γραμμών 457 και 462, ώστε το Dionaea να συνεργάζεται με την υπηρεσία VirusTotal και με το προγραμματάκι p0f.

Αποθηκεύουμε τις αλλαγές με [CTRL+O] και [Enter] κι εγκαταλείπουμε το nano με [CTRL+X].

Ενεργοποίηση της παγίδας

Αυτό ήταν, είμαστε έτοιμοι να ξεκινήσουμε τη λειτουργία του malware honeypot. Αρχικά, όμως, πρέπει να ξεκινήσουμε το p0f:

```
root@dionaea:~# p0f -i any -Q /tmp/p0f.sock -l -d -o /var/log/
p0f.log
p0f - passive os fingerprinting utility, version 2.0.8
```

```
(C) M. Zalewski <lcamtuf@dione.cc>, W. Stearns <wstearns@pobox.
com>
p0f: listening (SYN) on 'any', 262 sigs (14 generic, cksum
0F1F5CA2), rule: 'all'.
[*] Accepting queries at socket /tmp/p0f.sock (timeout: 2 s).
```

Τώρα εκκινούμε και το Dionaea, δίνοντας το ακόλουθο:

```
root@dionaea:~# /opt/dionaea/bin/dionaea -D -l all,-debug -L '*'
-p /opt/dionaea/var/run/dionaea.pid
Dionaea Version 0.1.0
Compiled on Linux/x86 at Mar 28 2012 21:39:25 with gcc 4.5.2
Started on dionaea running Linux/i686 release 2.6.32-
042stab044.11
```

Τα παραπάνω θα πρέπει να τα εκτελείται χειροκίνητα *κάθε* φορά που θέλετε να ξεκινήσετε τη λειτουργία του honeypot (π.χ., μετά από κάθε reboot). Αν θέλετε να επιβεβαιώσετε ότι τα p0f και dionaea τρέχουν σωστά, δώστε ένα "ps x" (πάντα χωρίς τα εισαγωγικά). Στην έξοδο του ps θα πρέπει να δείτε τα δύο προγράμματα στη λίστα με τις εκτελούμενες διεργασίες. Επίσης, δίνοντας "netstat –antpu", θα πρέπει να δείτε δύο γραμμές σαν τις ακόλουθες:

```
tcp  0  0    0.0.0.0:135     0.0.0.0:*LISTEN    3260/dionaea
tcp  0  0    0.0.0.0:42      0.0.0.0:*LISTEN    3260/dionaea
[...]
udp  0  0    0.0.0.0:5060    0.0.0.0:*LISTEN    3260/dionaea
```

Αν πήγαν όλα καλά, η παγίδα μας είναι έτοιμη να πιάσει malware που σουλατσάρουν ανεξέλεγκτα στο Διαδίκτυο! Γυρίστε λοιπόν τώρα στη σελίδα 90, για να δείτε πώς μπορούμε να χρησιμοποιούμε το Dionaea προκειμένου να βγάζουμε εξαιρετικά πολύτιμα συμπεράσματα για το τι κυκλοφορεί Εκεί Έξω (TM).

Το εργαλείο p0f και το Dionaea honeypot είναι εγκατεστημένα, ρυθμισμένα και ήδη σε υπηρεσία. Malware του κόσμου όλου, ελάτε κοντά!

```
   22 ?        S      0:00 [fsnotify_mark]
   23 ?        S      0:00 [ecryptfs-kthrea]
   24 ?        S<     0:00 [crypto]
   32 ?        S<     0:00 [kthrotld]
   34 ?        S      0:00 [scsi_eh_0]
   35 ?        S      0:00 [scsi_eh_1]
   36 ?        S      0:00 [kworker/u:3]
  213 ?        S      0:00 [scsi_eh_2]
  234 ?        S      0:00 [jbd2/sda1-8]
  235 ?        S<     0:00 [ext4-dio-unwrit]
  287 ?        S      0:00 upstart-udev-bridge --daemon
  291 ?        Ss     0:00 udevd --daemon
  479 ?        S<     0:00 [kpsmoused]
  561 ?        S      0:00 upstart-socket-bridge --daemon
  605 ?        Ss     0:00 /usr/sbin/sshd -D
  681 tty4     Ss+    0:00 /sbin/getty -8 38400 tty4
  685 tty5     Ss+    0:00 /sbin/getty -8 38400 tty5
  696 tty2     Ss+    0:00 /sbin/getty -8 38400 tty2
  697 tty3     Ss+    0:00 /sbin/getty -8 38400 tty3
  699 tty6     Ss+    0:00 /sbin/getty -8 38400 tty6
  700 ?        Ss     0:00 cron
  747 tty1     Ss+    0:00 /sbin/getty -8 38400 tty1
  882 ?        S      0:00 udevd --daemon
  883 ?        S      0:00 udevd --daemon
  899 ?        Ss     0:00 sshd: sub0 [priv]
 1024 pts/0    S      0:00 sudo su
 1025 pts/0    S      0:00 su
 1026 pts/0    S      0:00 bash
10386 ?        R      0:00 [kworker/0:0]
17823 ?        S      0:00 [flush-8:0]
17827 ?        Ss     0:00 p0f -i any -Q /tmp/p0f.sock -l -d -o /var/log/p0f.log
17848 ?        Ssl    0:00 /opt/dionaea/bin/dionaea -D -l all,-debug -L * -p /opt/dionaea/var/run/dionaea.pid
17849 ?        Ss     0:00 /opt/dionaea/bin/dionaea -D -l all,-debug -L * -p /opt/dionaea/var/run/dionaea.pid
17856 pts/0    R+     0:00 ps x
root@dionaea:~#
```

Skill: Intermediate

Tags: Wordlist, Dictionary, cupp, phonelist, mobilelist, birthdateslist, crunch, wyd, python, bash

Wordlist Generation: The Gossip Angle

by Dr@G



Τώρα που μάθαμε τα βασικά για την κατασκευή των wordlist, μπορούμε να κάνουμε το επόμενο βήμα: Να δημιουργήσουμε ειδικά λεξικά, προσαρμοσμένα στις ιδιαιτερότητες συγκεκριμένων στόχων. Με τέτοια προσωποποιημένα λεξικά, οι πιθανότητες επιτυχίας των bruteforce attacks είναι εξαιρετικά αυξημένες!

Καλώς ήρθατε στη συνέχεια του wordlist generation! Αυτή τη φορά θα δώσουμε έμφαση στην κατασκευή προσωποποιημένων λεξικών, σχεδιασμένων κατάλληλα για την αποτελεσματική επίθεση σε συγκεκριμένους ανθρώπους. Από το άρθρο μας στο deltaHacker 006, είδαμε ότι η παραγωγή αποτελεσματικών password βασίζεται σε πολλές παραμέτρους/λεπτομέρειες. Οι κάτοικοι κάθε χώρας έχουν τη δική τους γλώσσα, το δικό τους τρόπο ζωής και γενικότερα τις ιδιαιτερότητες τους. Κάνετε αν θέλετε μια μικρή έρευνα σε φίλους, συγγενείς και γνωστούς. Πάρτε το laptop ή το netbook μαζί σας και ζητείστε --διακριτικά και με τρόπο-- τον κωδικό για το ασύρματο δίκτυο του καθενός. Είμαστε βέβαιοι ότι θα εκπλαγείτε με το αποτέλεσμα της έρευνας: Οι περισσότεροι χρησιμοποιούν τον εργοστασιακό κωδικό, το σταθερό τους τηλέφωνο, το κινητό, το όνομα του/της συζύγου ή το όνομα κάποιου κατοικιδίου. Είναι προφανές, λοιπόν, ότι κάθε φορά που σκοπεύουμε να επιτεθούμε σε συγκεκριμένο στόχο --πάντα για εκπαιδευτικούς σκοπούς και με στόχο τη βελτίωση της ασφάλειάς του--, πρέπει να έχουμε ετοιμάσει ένα ειδικό wordlist που να είναι απόλυτα προσαρμοσμένο και προσωποποιημένο.

Για να φτιάξουμε προσαρμοσμένα και αποτελεσματικά λεξικά, πρέπει να μελετήσουμε το στόχο μας και να καταγράψουμε όποιες πληροφορίες καταφέρουμε να συγκεντρώσουμε. Για την ευκολότερη μελέτη της δράσης μας στο μέλλον, καλό θα ήταν να συγκεντρώνουμε τις πληροφορίες σε αρχεία απλού κειμένου. Τα ονόματα των οποίων, μάλιστα, καλό είναι να αντικατοπτρίζουν το στόχο, το μέρος αλλά και την περίοδο που έλαβε χώρα το λεγόμενο information gathering. Ένα τέτοιο όνομα θα μπορούσε να είναι το Annoula_AgiaParaskeui_21-01-2012.txt.

Λογικό είναι τώρα ν' αναρωτιέστε πώς θα κινηθούμε στο παρόν άρθρο. Αρχικά, θα κατασκευάσουμε διάφορα scripts σε Python και BASH, τα οποία παράγουν wordlists με βάση διάφορες ημερομηνίες ή τηλεφωνικούς αριθμούς. Μετά από τα scripts θα περάσουμε σε κάποια εργαλεία της διανομής Backtrack, τα οποία φημίζονται για την παραγωγή προσωποποιημένων λεξικών. Μιλάμε για τα εργαλεία cupp (common user passwords profiles), crunch και wyd (who's your daddy password profiler). Τα wordlist που θα κατασκευάσουμε θα μπορούν να αξιοποιηθούν σε επιθέσεις brute force.

Δεν λένε ψέματα, προδίδουν όμως!

Αλήθεια, έχει τύχει ποτέ να δείτε τον κωδικό κάποιου φίλου σας, την ώρα που τον πληκτρολογεί; Ακόμα κι αν δεν σας έχει τύχει κάτι τέτοιο, όλο και κάποια στιγμή θα μάθατε τον κωδικό ενός φίλου, επειδή χρειάστηκε να σας τον πει. Όπως καταλαβαίνετε, το θέμα μας έχει να κάνει με τους κωδικούς που επιλέγουν οι λιγότερο υποψιασμένοι χρήστες. Οι περισσότεροι από αυτούς εστιάζουν στους αριθμούς και χρησιμοποιούν για password διάφορες ημερομηνίες, όπως αυτή της γέννησής τους ή της επετείου της σχέσης τους κ.λπ. Αν νομίζετε ότι υπερβάλλουμε στο συγκεκριμένο θέμα, σκεφτείτε ότι οι περισσότερες επιθέσεις στο Facebook έχουν γίνει σε λογαριασμούς με αδύναμα και προβλέψιμα password. Φανταστείτε κάποιον, ο οποίος χρησιμοποιεί σαν κωδικό την ημερομηνία γέννησής του, την ίδια στιγμή που έχει δηλώσει τη συγκεκριμένη ημερομηνία στο προφίλ του! Πόσο δύσκολο θα ήταν για κάποιον να εκμεταλλευτεί αυτό το γεγονός;

Το πρώτο μας script είναι γραμμένο σε BASH. Η δουλειά του είναι να παράγει ένα string για κάθε ημερομηνία, από την 01/01/1950 ως την 31/12/2012. Τα string του script θα έχουν τη μορφή ddmmyyyy. Θα περιλαμβάνουν δηλαδή οκτώ ψηφία, θα ξεκινούν με την ημέρα, θα συνε-

χίζουν με το μήνα και θα καταλήγουν στο έτος. Με λίγα λόγια, το script θα καταγράψει όλες τις ημερομηνίες εντός του χρονικού διαστήματος το οποίο αναφέραμε, με τον τρόπο που τις διαβάζουμε στην Ελλάδα (πρώτα η μέρα, μετά ο μήνας και στο τέλος ο χρόνος). Για παράδειγμα, ένα από τα παραγόμενα string είναι το 21122012 -- η υποψήφια ημερομηνία για την καταστροφή του κόσμου :) Ας δούμε το script:

```
#!/bin/bash
#birthdates.sh by dr@g
for years in {1950..2012}
do
  for months in {1..12}
  do
    for days in {1..31}
      do
       printf "%02d%02d%04d\n" $days $months $years
      done
  done
done
```

Όπως βλέπετε, το προγραμματάκι μας αποτελείται από τρεις φωλιασμένους βρόχους (nested loops), οι οποίοι "σαρώνουν" όλα τα έτη, όλους τους μήνες και όλες τις ημέρες εντός της χρονικής περιοχής την οποία ορίζουμε. Φυσικά, αν γνωρίζουμε την ηλικία του υποψήφιου θύματός μας --έστω και στο περίπου--, μπορούμε να εκμεταλλευτούμε αυτή τη γνώση για να περιορίσουμε την "περιοχή εργασίας" του script. Με αυτό τον τρόπο θα πάρουμε ένα μικρότερο wordlist και η μετέπειτα επίθεσή μας θα διαρκέσει λιγότερο. Βέβαια, σε αυτήν την περίπτωση εκτός από τη διάρκεια της επίθεσης θα μειωθεί και η πιθανότητα επιτυχίας! Γενικά, πρέπει να είμαστε πολύ προσεκτικοί με τους συμβιβασμούς που κάνουμε, όταν δημιουργούμε ένα wordlist.

Αφού γράψουμε το script --μπορείτε να το κατεβάσετε και από το http://bit.ly/dh007birthdates--, το πρώτο πράγμα που πρέπει να κάνουμε είναι να το μετατρέψουμε σε εκτελέσιμο. Στη συνέχεια μπορούμε να το τρέξουμε, κατευθύνοντας την έξοδό του σε κάποιο αρχείο. Σε αυτό το αρχείο θα αποθηκευτεί το wordlist. Πρόκειται για δύο απλούστατα βήματα:

```
root@btvm:~# chmod +x birthdates.sh
root@btvm:~# ./birthdates.sh > birthdates.txt
```

Περισσότεροι αριθμοί, εξίσου προδοτικοί

Μια άλλη κατηγορία αριθμών που χρησιμοποιούνται συχνά ως passwords, είναι οι τηλεφωνικοί. Ο αριθμός του τηλεφώνου, είτε πρόκειται για κινητό είτε για σταθερό, αποτελεί κάτι που εύκολα μπορεί να θυμάται ο κάτοχός του. Ακριβώς γι' αυτό, ειδικά στις περιπτώσεις των wireless router, οι χρήστες που επιλέγουν ως κωδικό το τηλέφωνό τους είναι εντυπωσιακά πολλοί! Όπως πάντα, αν αποφασίσουμε να φτιάξουμε wordlist με τηλέφωνα θα πρέπει να αξιοποιήσουμε κάθε γνώση που διαθέτουμε για το υποψήφιο θύμα μας. Για παράδειγμα, αν γνωρίζουμε ότι μένει στο Ηράκλειο της Κρήτης, θα φτιάξουμε ένα wordlist με τηλεφωνικούς αριθμούς οι οποίοι ξεκινούν με το 2810. Συγκεκριμένα, στο wordlist θα καταγράψουμε όλους τους αριθμούς από το 2810000000 έως το 2810999999.

Στην Αθήνα --και συγκεκριμένα στου Ζωγράφου-- οι αριθμοί βρίσκονται στο διάστημα από 2107700000 έως 2107799999, ενώ στο κέντρο της πόλης βρίσκονται στην περιοχή από 2108200000 έως 2108299999. Κάπου εδώ, βέβαια, πρέπει ν' αναφέρουμε ότι η εισαγωγή νέων εταιρειών στο χώρο της τηλεφωνίας περιπλέκει κάπως τα πράγματα. Για παράδειγμα, η Vodafone στο Ηράκλειο της Κρήτης δίνει αριθμούς που ξεκινούν με το 2813 και είναι εντελώς διαφορετικοί από εκείνους του ΟΤΕ. Επομένως, θα ήταν ιδιαίτερα χρήσιμο να γνωρίζουμε την εταιρεία τηλεφωνίας, την οποία χρησιμοποιεί το υποψήφιο θύμα. Γενικά, όπως έχουμε πει επανειλημμένα, όσο πιο πολλά πράγματα ξέρουμε για το θύμα, τόσο το καλύτερο.

Προφανώς, τα παραπάνω ισχύουν και για τα κινητά τηλέφωνα. Αν γνωρίζουμε την εταιρεία τηλεφωνίας του στόχου μας, μπορούμε να περιορίσουμε το μέγεθος της λίστας *χωρίς* να πλήξουμε την αποτελεσματικότητά της. Για παράδειγμα, η CosmOTE χρησιμοποιεί αριθμούς της μορφής 6972xxxxxx. Βέβαια, με τα κινητά τηλέφωνα δεν είναι και τόσο απλά τα πράγματα. Μπαίνει στη μέση, βλέπετε, τη ζήτημα της φορητότητας των αριθμών. Ακριβώς γι' αυτό, αν ο στόχος μας είχε περάσει και από άλλες εταιρείες τηλεφωνίας, θα ήταν χρήσιμο να το γνωρίζουμε. Ο χρυσός κανόνας της συλλογής πληροφοριών επιβεβαιώνεται και πάλι!

Στη συνέχεια βλέπουμε δύο σύντομα scripts σε Python, τα οποία παράγουν wordlists με τα τηλέφωνα της περιοχής Ζωγράφου, καθώς και με εκείνα της CosmOTE:

```
#! /usr/bin/env python
# zografou land-line phone numbers
for i in range(2107700000,2107799999):
    print i
#! /usr/bin/env python
# cosmote mobile phone numbers 6972xxxxxx
for z in range(6972000000,6972999999):
    print z
```

Όπως βλέπετε, κάθε script αποτελείται από ένα απλό βρόχο for. Με τη βοήθειά του τυπώνουμε όλους τους αριθμούς στην περιοχή που μας ενδιαφέρει. Αφούς πληκτρολογήσετε τα script σε κάποιον text editor και τ' αποθηκεύσετε, θυμηθείτε να τα κάνετε εκτελέσιμα. Παράδειγμα:

```
root@btvm:~# chmod +x zografou_phones.py 6972_mobiles.py
```

Στη συνέχεια μπορείτε να τα εκτελέσετε, ανακατευθύνοντας τις εξόδους τους σε αρχεία:

```
root@btvm:~# ./zografou_phones.py > zografou_phones.txt
root@btvm:~# ./6972_mobiles.py > 6972_mobiles.txt
```

Τέλεια! Τα αρχεία zografou_phones.txt και cosmote_6972.txt περιλαμβάνουν πλέον τα αντίστοιχα wordlist. Με αυτό το μικρό οπλοστάσιο στα χέρια μας, μπορούμε να περάσουμε σε μερικά όπλα (πυρηνικά!) του BackTrack Linux.

Common User Passwords Profiles [cupp.py]

Ας μιλήσουμε υποθετικά: Μια όμορφη γειτόνισσα μένει πάνω από το ταπεινό μου διαμέρισμα, έχει μια πολύ γρήγορη σύνδεση και το ασύρματο

ρουτεράκι της εκπέμπει εξαιρετικό σήμα. Όμως η μόνη δραστηριότητά της στο Internet σχετίζεται με το Facebook. Δεν είναι κρίμα να μένει ανεκμετάλλευτο τόσο bandwidth; Δεν μιλάμε για ζημιές, καταστροφές ή κάποιου είδους παρενόχληση. Μόνο για την αξιοποίηση των ανεκμετάλλευτων πόρων! Κάπως έτσι, αποφάσισα να βρω τον κωδικό του ασύρματου δικτύου της. Ναι, τελικά δεν σας μιλάω καθόλου υποθετικά :)

Το όνομα του δικτύου της είναι: Thomson2A5570. Όπως ξέρουμε από το σχετικό άρθρο του deltaHacker 001, ο αλγόριθμος παραγωγής κωδικών στα router της Thomson λαμβάνει υπόψη του τα τελευταία έξι ψηφία του SSID. Στην περίπτωσή μας, δηλαδή, τα “2A5570”. Χρησιμοποιώντας το εργαλείο ReverseWiFicracker.exe, από το ίδιο τεύχους του περιοδικού, βρήκαμε εύκολα τον αντίστοιχο κωδικό: 53057CDE9D. Δυστυχώς, όμως, το συγκεκριμένο password δεν ήταν σωστό! Η όμορφη γειτόνισσα είχε αντικαταστήσει το εργοστασιακό password με κάποιο δικό της. Όπως καταλαβαίνετε, αυτό το εμπόδιο δεν ήταν ικανό να με ωθήσει σε παραίτηση. Δεν θα το έβαζα κάτω τόσο εύκολα! Έτσι, επιστράτευσα όσες τεχνικές κατέχω στο social engineering, για να μαζέψω όσο το δυνατόν περισσότερες πληροφορίες...

Το όνομα του στόχου είναι Κατερίνα Αδάμου, αλλά έχω ακούσει τον άντρα της να τη φωνάζει χαϊδευτικά Κατέ. Παρεμπιπτόντως, το όνομα του άντρα της είναι Κώστας. Τις προάλλες, όμως, όταν γύριζαν με τα ψώνια από το super market, την άκουσα να φωνάζει "Μπούμπηηηηη, έλα μωρό μου να πάρεις τις τσάντες". Περιττό να πω ότι ο Μπούμπης προστέθηκε άμεσα στο αρχείο με τις πληροφορίες μου! (Σ.τ.Ε. ΟΚ, δεν μιλούσα τόση ώρα, περιττό όμως να σημειώσω ότι ανησυχώ :S)

Σε μια συνάντηση για τα κοινόχρηστα της πολυκατοικίας, της είπα ότι οι περισσότεροι άνθρωποι --και καλά-- έχουν γενέθλια σε συγκεκριμένους μήνες και συνήθως προς το μέσο τους. Με την κουβεντούλα που ξεκίνησε, βρήκα την αφορμή να ρωτήσω πότε γεννήθηκε η ίδια. Εντελώς αθώα, μου έδωσε την εξής ημερομηνία: 28/9/1984. Φυσικά, στην ίδια συνάντηση δεν παρέλειψα να ρωτήσω και την ημερομηνία γέννησης του άνδρα της, με τη δικαιολογία ότι αυτή μοιάζει πολύ μικρότερή του. Ναι, το χειρίστηκα πονηρά το θέμα, αλλά έμαθα την ημερομηνία γέννησης και των δύο! Συνεχίζοντας το information gathering, έμαθα ότι το παιδί τους ονομάζεται Δημήτρης κι ότι το φωνάζουν Μήτσο ή Μητσάρα! Φυσικά, φρόντισα να μάθω και την ημερομηνία γέννησης του Μητσάρα. Να μην παραλείψω ότι έχουν τρέλα με τα ζώα. Συγκεκριμένα, έχουν ένα σκυλάκι, τον Αζόρ. Η δουλειά του άντρα της είναι τον ΟΤΕ (τον βλέπω καθημερινά να αφήνει το υπηρεσιακό αυτοκίνητο, έξω από την πολυκατοικία). Όπως αντιλαμβάνεστε, κάθε λεπτομέρεια μετρούσε και φρόντισα να μάθω όσο πιο πολλά μπορούσα. Έμαθα, λοιπόν, ότι ο τύπος είναι φανατικός Ολυμπιακός. Μάλιστα κάθε Κυριακή ντύνεται στα κόκκινα, ντύνει το παιδί του σαν τον Μιραλά και πάνε μαζί στο γήπεδο! Τέλος, να μην ξεχάσω να σας πω ότι η καταγωγή τους είναι από Ικαρία κι από Καλαμάτα.

Πολύ ωραία όλα αυτά, θα πείτε, αλλά πώς θα μας φανούν χρήσιμα; Όπως έχουμε ήδη πει, είναι μάλλον δύσκολο να μπούμε στη σκέψη και στη φιλοσοφία κάθε ανθρώπου. Έτσι, προσπαθούμε να αντλήσουμε όσο το δυνατόν περισσότερες πληροφορίες από την καθημερινή του ζωή, ώστε

να φτιάξουμε μια προσωποποιημένη λίστα πιθανών κωδικών. Ας επιστρέψουμε όμως στην Κατερίνα και στο ασύρματο ρουτεράκι της. Αφού είχα συγκεντρώσει τις πληροφορίες που σας ανέφερα, αποφάσισα να αξιοποιήσω ένα Python script από το BackTrack. Αναφέρομαι στο cupp.py. Αυτό το φοβερό και τρομερό πρόγραμμα παράγει πανίσχυρα wordlist με μεγάλη ευκολία, αξιοποιώντας όσες προσωπικές πληροφορίες έχουμε συλλέξει για το θύμα μας. Δείτε πώς το χρησιμοποίησα για το ασύρματο δίκτυο της Κατερίνας:

```
root@btvm:~# cd /pentest/passwords/cupp
root@btvm:/pentest/passwords/cupp# python cupp.py -i
[+] Insert the informations about the victim to make a dictionary
[+] If you don't know all the info, just hit enter when asked! ;)
> Name: Katerina
> Surname: Adamou
> Nickname: Kate
> Birthdate (DDMMYYYY): 28091984
> Wife's(husband's) name: Kostas
> Wife's(husband's) nickname: Boubis
> Wife's(husband's) birthdate (DDMMYYYY): 19111976
> Child's name: Dimitris
> Child's nickname: Mitsos
> Child's birthdate (DDMMYYYY): 16052005
> Pet's name: Azor
> Company name: OTE
> Do you want to add some key words about the victim? Y/[N]: Y
> Please enter the words, separated by comma. [i.e. hacker,
juice, black]: Olympiakos,Thrylos,Thrilos,Ikaria,Kalamata
> Do you want to add special chars at the end of words? Y/[N]: N
> Do you want to add some random numbers at the end of words? Y/
[N]Y
> Leet mode? (i.e. leet = 1337) Y/[N]: N
[+] Now making a dictionary...
[+] Sorting list and removing duplicates...
[+] Saving dictionary to katerina.txt, counting 4649 words.
[+] Now load your pistolero with katerina.txt and shoot! Good
luck!
root@btvm:/pentest/passwords/cupp#
```

Όπως βλέπετε, έτρεξα το cupp με την παράμετρο i και ξεκίνησε στο λεγόμενο interactive mode. Έτσι, το script άρχισε να μου απευθύνει διάφορες ερωτήσεις για το στόχο, όπως και κάποιες γενικότερες, για τον τρόπο λειτουργίας του. Για παράδειγμα, στο τέλος με ρώτησε αν θέλω να ενεργοποιηθεί το "leet mode". Σε αυτό το mode, παράγονται και κωδικοί στους οποίους ορισμένα γράμματα έχουν αντικατασταθεί από αριθμούς! Όταν απάντησα σ' όλες τις ερωτήσεις, το πρόγραμμα δημιούργησε μια woldlist με 4649 λέξεις και την αποθήκευσε στο αρχείο katerina.txt. Εγώ συνδύασα αυτή τη wordlist με το aircrack-ng, δίνοντας κάτι τέτοιο:

```
root@btvm:~# aircrack-ng Thomson2A5570-01.cap -w ~/katerina.txt
```

Περιττό να σας πω ότι τα κατάφερα στο πι και φι! Ο κωδικός που χρησιμο-

ποιούσε η γειτόνισσα ήταν Katerina84.

Wordlist με το Crunch

Το Crunch είναι ακόμη ένα σημαντικό εργαλείο για παραγωγή wordlists. Πρόκειται για ένα ιδιαίτερα παραμετροποιήσιμο πρόγραμμα που επιτρέπει να δημιουργούμε εξαιρετικά προσαρμοσμένες wordlists. Για παράδειγμα, αν θέλουμε passwords με μικρά γράμματα (lower case) και αριθμούς (numerics) με μήκος από έναν έως τέσσερις χαρακτήρες, αρκεί να δώσουμε κάτι τέτοιο:

```
root@btvm:~# cd /pentest/passwords/crunch

root@btvm:/pentest/passwords/crunch# ./crunch 1 4 -f charset.lst
lalpha-numeric -o wordlist.lst

Crunch will now generate the following amount of data: 8588664
bytes

8 MB

0 GB

0 TB

0 PB

Crunch will now generate the following number of lines: 1727604
100%

root@btvm:/pentest/passwords/crunch#
```

Όπως καταλαβαίνετε, η λίστα θα αποθηκευτεί στο αρχείο wordlist.lst. Στα ασύρματα δίκτυα, χρειαζόμαστε συνήθως μια wordlist με όλους τους οκταψήφιους δεκαεξαδικούς αριθμούς. Το Crunch παράγει τέτοιες λίστες σε μια γραμμή:

```
root@btvm:/pentest/passwords/crunch# ./crunch 8 8 -f charset.lst
hex-upper -o hex_wordlist.lst
```

Βέβαια, αν επιχειρήσετε κάτι τέτοιο, θα χρειαστείτε *αρκετή* υπομονή. Η λίστα που θα προκύψει θα έχει μέγεθος περί τα 36GB! Αν σπάσουν τα νεύρα σας και διακόψετε τη διαδικασία, έχετε κατά νου ότι χρησιμοποιώντας την παράμετρο –s μπορείτε να πείτε στο Crunch να συνεχίσει από εκεί που είχε σταματήσει προηγουμένως. Έτσι, αν ξεκινήσατε την παραγωγή των οκταψήφιων δεκαεξαδικών και πατήσατε κάποια στιγμή το [CTRL+C], μπορείτε να συνεχίσετε τη διαδικασία δίνοντας αυτό:

```
root@btvm:/pentest/passwords/crunch# ./crunch 8 8 -f charset.lst
hex-upper -o hex_wordlist.lst -s
```

Ένα ακόμα προσόν του Crunch αφορά στην υποστήριξη οποιουδήποτε charset θέλουμε. Για παράδειγμα, μπορούμε να δώσουμε κάτι τέτοιο:

```
root@btvm:/pentest/passwords/crunch# ./crunch 4 4
abcABCαβγΑΒΓ123!# > mix_char_set.lst
```

Έτσι, το Crunch θα χρησιμοποιήσει μόνον τους χαρακτήρες που δηλώσαμε. Μια χρήσιμη παράμετρος του προγράμματος είναι και η –t. Αν γνωρίζουμε ότι το συνθηματικό που ψάχνουμε ξεκινά μ' έναν συγκεκριμένο αριθμό και μια παύλα, ενώ συνεχίζει με έναν τετραψήφιο αριθμό στο δεκαδικό, μπορούμε να κατασκευάσουμε μια προσαρμοσμένη wordlist. Κάτι τέτοιο θα γινόταν ως εξής:

```
root@btvm:/pentest/passwords/crunch# ./crunch 17 17 0123456789 -t
00AA1E69BCFE-@@@@ > passwords.txt
```

Παρεμπιπτόντως, η λίστα που μόλις φτιάξαμε είναι ιδανική για bruteforce

attacks σε κάποια ασύρματα δίκτυα συγκεκριμένης εταιρείας. Το όνομά της σας το αφήνουμε ως άσκηση :D

Who's Your Daddy Password Profiler [wyd.pl]

Πολλές φορές δεν θέλουμε να δημιουργήσουμε μια wordlist από ένα σύνολο χαρακτήρων, αλλά από ένα σύνολο αφηρημένων πληροφοριών. Για παράδειγμα, μπορεί να μην ξέρουμε τίποτα για την προσωπική ζωή του στόχου μας, αλλά να ξέρουμε ότι επισκέπτεται συχνά ένα site, ότι του αρέσει πολύ ένα τραγούδι κ.λπ. Ε, λοιπόν, σ' αυτή την περίπτωση το εργαλείο wyd.pl (σε Perl) μπορεί ν' αποδειχτεί πολύτιμος βοηθός! Αρκεί να σας πούμε ότι υποστηρίζει αρχεία Plaintext, HTML/PHP, DOC/PPT, MP3, JPG και ODT/ODS/ODP. Το μόνο που έχουμε να κάνουμε εμείς είναι να του δώσουμε το σχετικό υλικό που έχουμε συγκεντρώσει! Το wyd.pl μπορούμε να το κατεβάσουμε από το http://packetstormsecurity.org/files/54601/wyd-0.2.tar.gz.html. Αν θέλουμε να ενεργοποιήσουμε όλες τις δυνατότητές του θα χρειαστεί να εγκαταστήσουμε στο σύστημά μας ορισμένα πρόσθετα πακέτα. Για παράδειγμα, για την υποστήριξη αρχείων DOC/PPT θα χρειαστούμε το catdoc, για αρχεία MP3 το mp3info και για αρχεία JPG το jhead.

Ας περάσουμε τώρα σ' ένα σενάριο χρήσης. Ας υποθέσουμε ότι γνωρίζουμε το site που επισκέπτεται συχνά ο στόχος μας. Αφού κατεβάσουμε κι αποσυμπιέσουμε το πακέτο με το wyd.pl, φτιάχνουμε έναν κατάλογο μέσα στον οποίο θα κατεβάσουμε τα περιεχόμενα του site. Στη συνέχεια, αρκεί να τρέξουμε το wyd.pl δίνοντάς του ως παράμετρο το αρχείο στο οποίο θα αποθηκευτεί το wordlist, καθώς και τον κατάλογο με τα αρχεία του site. Εν ολίγοις, αν υποθέσουμε ότι ο στόχος μας διατηρεί --ή τέλος πάντων επισκέπτεται συχνά-- το www.target_site.gr, μπορούμε να δώσουμε τα ακόλουθα:

```
root@btvm:~# mkdir ~/target_site
root@btvm:~# cd ~/target_site
root@btvm:~# wget -r http://www.target_site.gr
root@btvm:~# perl wyd.pl -o wordlist -t -b -e ~/target_site
```

Όπως είπαμε, στο wyd.pl μπορούμε να δώσουμε και μεμονωμένα αρχεία, διαφόρων τύπων. Για να κατανοήσετε τη μεγάλη αξία του συγκεκριμένου εργαλείου, σκεφτείτε πόση δουλειά θα χρειαζόταν από μέρους σας για να φιλτράρετε τις πιθανά χρήσιμες πληροφορίες από διάφορα έγγραφα ή κάποιο site.

Μερικά συμπεράσματα

Η παραγωγή καλών wordlists αποτελεί σημαντικό κομμάτι των επιθέσεων brute force και θέλουμε να πιστεύουμε ότι καλύψαμε όλες τις βασικές πτυχές της σχετικής διαδικασίας. Αξίζει ν' αναφέρουμε ότι σ' αυτήν την εργασία η καλή γνώση των regular expressions, καθώς κι εργαλείων όπως το sed και το awk, ενδέχεται να βοηθήσει σε μεγάλο βαθμό. Φυσικά, τα δικά μας script αλλά κι αυτά του BackTrack μπορούν να βοηθήσουν εξίσου πολύ. Πάντως το social engineering και η δημιουργική φαντασία του καθενός αποτελούν τα σημαντικότερα εργαλεία! Από εκεί και πέρα, σε άρθρο που θα δημοσιεύσουμε πολύ σύντομα θα δούμε αναλυτικά πώς χρησιμοποιούνται οι wordlists που φτιάχνουμε για το σπάσιμο κωδικών router, την αντιστροφή MD5 hashes, τη σύνδεση σε FTP servers και πολλά άλλα.

Μείνετε συντονισμένοι!

Το deltaHacker είναι το **μοναδικό στο χώρο των περιοδικών** έντυπο που **ασχολείται αποκλειστικά κι αληθινά** με τον κόσμο του **ethical hacking**, γυρίζοντας συστηματικά την πλάτη του στον καθωσπρεπισμό κι αδιαφορώντας παντελώς για την ασφάλεια της θεωρίας.

Το deltaHacker είναι τόσο σίγουρο για τον εαυτό του και τόσο τίμιο κι ευθύ απέναντι στον αναγνώστη, που αντί να γυρνάει εδώ κι εκεί στα περίπτερα και ν' αγωνίζεται να φανεί ανάμεσα στις Σούπερ-Κατίνες και στα PC-something-bored, **έρχεται κατευθείαν στο σπίτι σας** – κι έρχεται για να σας κατακτήσει.

Αν είστε από εκείνους που δεν βλέπουν τον υπολογιστή τους ως άλλη μια οικιακή συσκευή και δεν αντιμετωπίζετε το Internet ως ένα δίκτυο που αρχίζει και τελειώνει με το Facebook και το Twitter, τότε **είναι 102% βέβαιο ότι θ' αγαπήσετε το deltaHacker**, καθώς κι ότι θα θέλετε την παρέα του και τους 12 μήνες του χρόνου!

Όλες οι πληροφορίες για όλα τα προγράμματα συνδρομών είναι στο

deltahacker.gr/subscriptions

Παραγγείλετε τώρα μία από τις συμφέρουσες συνδρομές από το

deltahacker.gr/order

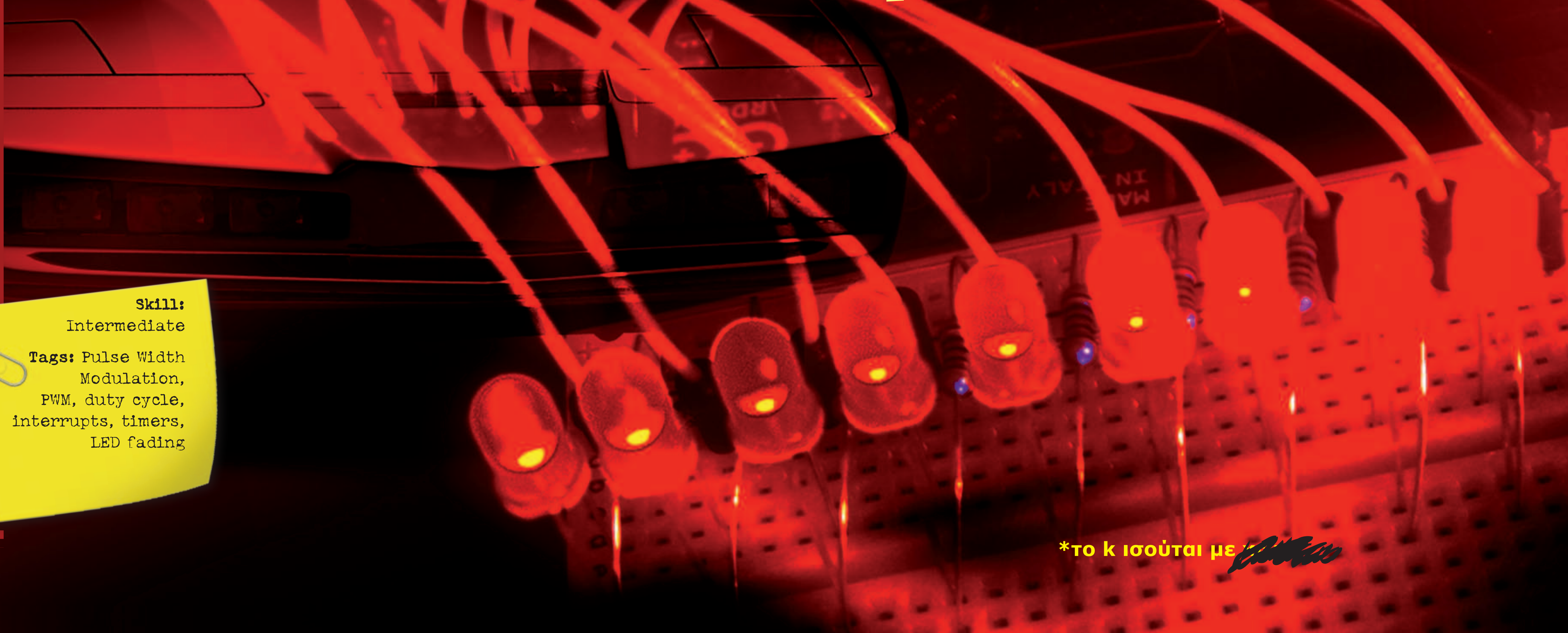
deltaHacker: Hacking for the rest of us.

Arduino

the way of the hacker!! (Μέρος 5/k*)

by Spir@lEvolution

Τι θα λέγατε για ένα Larson scanner με δέκα LED; Μιλάμε για μια συστοιχία LED, τα οποία αναβοσβήνουν με τέτοιον τρόπο ώστε να θυμίζουν το μάτι ενός Cylon -- ή τέλος πάντων το φωτεινό σηματάκι στην πρόσοψη του KITT. Όπως αντιλαμβάνεστε, θα εγκαταλείψουμε προσωρινά το RGB matrix και τους shift register για ν' ασχοληθούμε μ' ένα απλούστερο κύκλωμα. Στόχος μας είναι η γνωριμία με την τεχνική PWM και η υλοποίησή της σε software. Νομίζετε ότι κάτι τέτοιο είναι περιττό; Τι θα λέγατε αν δημιουργούσαμε περισσότερα κανάλια PWM, από αυτά που προσφέρει το hardware του Arduino; Με την τεχνογνωσία που θ' αποκτήσουμε θα μπορούμε αργότερα να φτιάξουμε παιχνίδια για το RGB matrix με εκατομμύρια *κυριολεκτικά* χρώματα! Εξάλλου, θα βρούμε και μια πρώτης τάξεως αφορμή, για να παίξουμε λίγο παραπάνω με τους timers και τα interrupts...



Skill: Intermediate

Tags: Pulse Width Modulation, PWM, duty cycle, interrupts, timers, LED fading

*το k ισούται με [redacted]

Η τελευταία μας κατασκευή στηριζόταν σε ένα RGB matrix. Θεωρητικά, κάθε κουκκίδα σε μια τέτοια διάταξη μπορεί να προβάλει *οποιαδήποτε* απόχρωση. Ωστόσο, το frame buffer του προγράμματός μας διατηρούσε ένα μόλις bit για κάθε χρωματική συνιστώσα (Red, Green και Blue) κάθε κουκκίδας. Με άλλα λόγια, για κάθε κουκκίδα το πρόγραμμα διατηρούσε 3bit πληροφορίας. Επομένως το πρόγραμμα διέθετε μια παλέτα των οκτώ μόλις χρωμάτων (βάθος χρώματος 3bit, 2^3=8 διαφορετικά χρώματα). Τα οκτώ χρώματα μπορεί να ακούγονται αρκετά, αλλά η απουσία των ενδιάμεσων αποχρώσεων καθιστά αδύνατη την υλοποίηση πολλών εφέ (fade-in, fade-out, morphing, κ.λπ). Για να ξεπεράσουμε αυτόν τον περιορισμό θα μπορούσαμε να κατασκευάσουμε ένα μεγαλύτερο frame buffer, το οποίο θα διατηρούσε περισσότερα bit πληροφορίας για το χρώμα κάθε κουκίδας. Όμως το αυξημένο πλήθος των bit δεν θα επαρκούσε. Θα χρειαζόμασταν κι έναν ειδικό μηχανισμό, ο οποίος θα μπορούσε να ρυθμίζει την ένταση κάθε χρώματος και να δημιουργεί τελικά τις επιθυμητές αποχρώσεις. Πώς όμως μπορεί να γίνει κάτι τέτοιο; Πώς μπορούμε να μεταβάλλουμε τη φωτεινότητα ενός LED, από τον κώδικα; Είμαστε σίγουροι ότι σκέφτεστε το PWM και τη σχετική συνάρτηση του Arduino, την analogWrite. Καλά κάνετε. Ωστόσο το Arduino διαθέτει μόνο έξι κανάλια PWM[1], ενώ τα matrix που γνωρίσαμε διαθέτουν γραμμές και στήλες των οκτώ LED. Επομένως, για την οδήγηση ενός τέτοιου matrix θα χρειαζόμασταν τουλάχιστον οκτώ κανάλια PWM. Φυσικά, θα χρειαζόμασταν κι ένα εντελώς διαφορετικό κύκλωμα, σε σχέση με όσα είδαμε. Θα χρειαζόμασταν ένα κύκλωμα στο οποίο το matrix θα συνδεόταν απευθείας πάνω στους ακροδέκτες του Arduino, ώστε να είναι δυνατή η χρήση των ενσωματωμένων καναλιών PWM. Όπως καταλαβαίνετε, αυτή η "λύση" είναι γεμάτη προβλήματα και σύντομα οδηγεί σε νέο αδιέξοδο. Αναρωτιέστε τι θα μπορούσαμε να κάνουμε για όλα αυτά; Η λύση, θεωρητικά τουλάχιστον, είναι πολύ απλή: Θα κατασκευάσουμε σε software το δικό μας σύστημα PWM, το οποίο θα διαθέτει οκτώ κανάλια και θα μπορεί να ρυθμίζει την τάση σε οποιουσδήποτε οκτώ ακροδέκτες -- ακόμα και μέσω του PSI και μερικών shift registers!

Κάτι για το τέλος, από τώρα

Κάπου εδώ θα μπορούσαμε να ξεκινήσουμε την ανάλυση περί PWM και μετά από μερικές παραγράφους ξερής θεωρίας, να σας χαιρετήσουμε! Μια τέτοια παρουσίαση όμως θα ήταν κουραστική και θα έμοιαζε εντελώς άσκοπη. Έτσι, αποφασίσαμε να συνδυάσουμε τη γνωριμία μας με την τεχνική PWM, με μια παραδοσιακή κατασκευή: Πρόκειται για ένα larson scanner, το σύστημα που υλοποιούν *όλοι* όσοι ασχολούνται με τα ηλεκτρονικά και τον προγραμματισμό, σε κάποια φάση της σταδιοδρομίας τους. Συγκεκριμένα, θα φτιάξουμε ένα larson scanner των δέκα LED, το οποίο θα προσομοιώνει το φωτεινό σηματάκι του KITT αλλά και το μάτι των Cylon[2]. Ελπίζουμε να γνωρίζετε και τον KITT (το αυτοκίνητο στον Ιππότη της Ασφάλτου) και τους Cylon (τα μοχθηρά ρομπότ στο BattleStar Galactica). Αν όχι και μέχρι να δείτε τις δύο σειρές, μπορείτε να ρίξετε μια ματιά στο βίντεο από τη δική μας κατασκευή:

`http://deltahacker.gr/?p=5037`

1. Το Arduino Mega διαθέτει περισσότερα κανάλια PWM, αλλά εμείς θα βρούμε μια λύση για το Arduino του λαού: Το μικρό Arduino Uno και τα συμβατά με αυτό!

2. Ναι, πρόκειται για διαφορετικά εφέ! Το σηματάκι του KITT ήταν μια κουκκίδα που πραγματοποιούσε παλινδρομική κίνηση και η οποία άφηνε πίσω της μια ουρίτσα. Στο μάτι των Cylon, η φωτεινή κουκκίδα πραγματοποιούσε και πάλι μια παλινδρομική κίνηση, αλλά εμφάνιζε δύο ουρίτσες: Μία πίσω της και μία μπροστά. Ξέρουμε... Ήταν πολύ καμμένη αυτή η πληροφορία ;-)

Προφανώς, εφόσον θα παίξουμε με δέκα LED, τα οποία θ' ανάβουν και θα σβήνουν ομαλά, το πρόγραμμα θα πρέπει να υλοποιεί ένα σύστημα PWM των δέκα καναλιών (ένα για κάθε LED). Θα περίμενε κανείς ότι πριν κατασκευάσουμε ένα τόσο ευρύ σύστημα PWM --ή έστω ένα των οκτώ καναλιών, το οποίο θα ήταν κατάλληλο και για την οδήγηση ενός matrix-- θα ξεκινούσαμε με κάτι μικρότερο. Ωστόσο, αυτή η προσέγγιση δεν θα μας προσέφερε τίποτα. Βλέπετε, η δυσκολία στην υλοποίηση ενός συστήματος PWM δεν έγκειται στο πλήθος των καναλιών, αλλά στην κατανόηση του ίδιου του PWM. Μόλις ξεκαθαρίσει κανείς τη σχετική αρχή λειτουργίας, μπορεί να κατασκευάσει εξίσου εύκολα ένα σύστημα του ενός καναλιού κι ένα των δεκαπέντε! Εξάλλου, δέκα κανάλια PWM και δέκα LED έχουν περισσότερη πλάκα από οκτώ ή και λιγότερα.

Βασικές γνώσεις

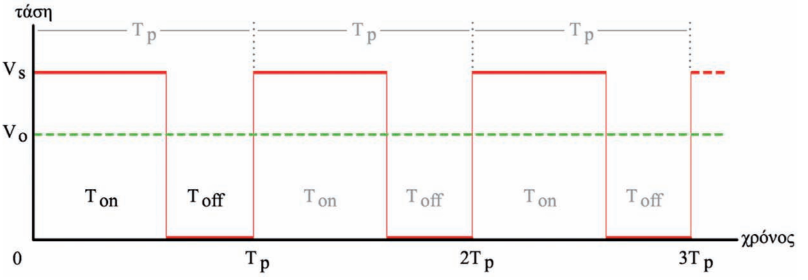
Η συνάρτηση analogWrite βρίσκεται στις βασικές βιβλιοθήκες του Arduino, είναι δε γνωστή ακόμα και στους νεοφερμένους. Αυτό που ενδέχεται να μη συνειδητοποιούν οι προαναφερθέντες είναι ότι η εν λόγω συνάρτηση στηρίζεται στο σύστημα PWM του Arduino. Ας δούμε, λοιπόν, τι ισχύει. Η συνάρτηση analogWrite επιτρέπει τη δημιουργία οποιασδήποτε τάσης από 0 έως 5V, επάνω σε ορισμένους ακροδέκτες. Σε γενικές γραμμές, αυτοί οι ακροδέκτες δεν έχουν καμία ιδιαιτερότητα: Μπορούν να βρίσκονται στη μία ή στην άλλη λογική κατάσταση (low ή high) και να εμφανίζουν την αντίστοιχη τάση (0 ή 5Volt). Μη φανταστείτε δηλαδή ότι μιλάμε για ειδικούς ακροδέκτες, ούτε ότι υπάρχει κάποιο αναλογικό κύκλωμα μέσα στο Arduino που συνθέτει τάσεις κατά παραγγελία. Όταν χρησιμοποιούμε τη συνάρτηση analogWrite η κατάσταση του επιλεγμένου ακροδέκτη εναλλάσσεται ταχύτατα και με κατάλληλο τρόπο, από low σε high και τούμπαλιν. Ως αποτέλεσμα αυτής της "κατάλληλης" εναλλαγής, ο μέσος όρος της τάσης του ακροδέκτη προσεγγίζει την τάση που έχουμε ζητήσει από την analogWrite! Σας φαίνεται περίπλοκο; Η αλήθεια είναι ότι ακούγεται περισσότερο σύνθετο απ' όσο είναι στην πραγματικότητα. Πριν όμως εμβαθύνουμε στο θέμα αξίζει να ξεκαθαρίσουμε μερικές ακόμα λεπτομέρειες. Ποιος εναλλάσσει "ταχύτατα και με τον κατάλληλο τρόπο" την κατάσταση του ακροδέκτη; Σε ποιους ακροδέκτες μπορεί να εφαρμοστεί η συνάρτηση analogWrite;

Καταρχάς, η εναλλαγή της κατάστασης του ακροδέκτη πραγματοποιείται από το σύστημα PWM. Με άλλα λόγια, η συνάρτηση analogWrite αποτελεί ένα interface, το οποίο διευκολύνει τον χειρισμό του PWM. Το συγκεκριμένο σύστημα δεν αποτελεί κάποιο αυτόνομο περιφερειακό, ξεκομμένο κι ανεξάρτητο από τα υπόλοιπα. Το κύκλωμα PWM αποτελεί υποσύστημα των timer. Συγκεκριμένα, καθένας από τους τρεις timer του Arduino διαθέτει δύο κανάλια PWM. Έτσι, προκύπτει και το μάλλον παράξενο πλήθος των διαθέσιμων καναλιών (έξι). Κάθε κανάλι συνδέεται σ' έναν συγκεκριμένο ακροδέκτη του Arduino και μπορεί να επιδράσει *μόνο* σ' αυτόν. Ως εκ τούτου, η συνάρτηση analogWrite μπορεί να εφαρμοστεί στους έξι ακροδέκτες, οι οποίοι συνδέονται στα έξι κανάλια PWM. Στην πλακέτα του Arduino, οι σχετικοί ακροδέκτες σημειώνονται με το χαρακτήρα "~".

Ποσοτική περιγραφή

Σύμφωνα με τα παραπάνω, η τεχνική PWM προβλέπει την κατάλληλη εναλλαγή της κατάστασης ενός ακροδέκτη, ώστε ο μέσος όρος της τάσης του να προσεγγίζει την επιθυμητή. Αυτή η περιγραφή δεν περιέχει κανένα σφάλμα, αλλά δεν διευκρινίζει απολύτως τίποτα. Με αυτή και μόνο

θα ήταν εξαιρετικά δύσκολο να κατασκευάσουμε το δικό μας σύστημα PWM. Ακριβώς γι' αυτό, θα σταματήσουμε τις αφηρημένες περιγραφές και θα εξετάσουμε το PWM στην πράξη. Στην εικόνα 1, η κόκκινη γραμμή αναπαριστά την τάση ενός ακροδέκτη πάνω στον οποίο εφαρμόζουμε την τεχνική PWM. Η στάθμη Vs αποτελεί την τάση της λογικής κατάστασης high και στην περίπτωσή μας είναι τα 5V. Η πράσινη γραμμή αναπαριστά το μέσο όρο της τάσης του ακροδέκτη, τον οποίο συμβολίζουμε με Vo. Με άλλα λόγια, το Vo αποτελεί την τάση που εμφανίζει ο ακροδέκτης εξαιτίας της εφαρμογής του PWM. Ας μελετήσουμε λίγο παραπάνω αυτό το διάγραμμα...



Εικόνα 1
Η κόκκινη γραμμή αναπαριστά την τάση ενός ακροδέκτη, επάνω στον οποίο εφαρμόζουμε την τεχνική PWM. Το Vs αναπαριστά την τάση που αντιστοιχεί στη λογική κατάσταση high (5V). Η πράσινη γραμμή αναπαριστά τη μέση τιμή της τάσης. Η τεχνική PWM επιβάλλει την εναλλαγή της κατάστασης του ακροδέκτη με σταθερή συχνότητα. Ωστόσο, η χρονική διάρκεια της κατάστασης high είναι διαφορετική από εκείνη της κατάστασης low.

Καταρχάς, είναι σαφές ότι κάθε στιγμή η τάση στον ακροδέκτη είναι 0 ή 5V. Με άλλα λόγια, ο ακροδέκτης βρίσκεται πάντα σε μία από τις δύο λογικές καταστάσεις (low ή high). Αναμενόμενο, θα σκεφτεί κανείς, αφού έχουμε ξεκαθαρίσει ότι πρόκειται για ακροδέκτη με ψηφιακή λειτουργία. Αυτό που δεν έχουμε ξεκαθαρίσει αφορά στον τρόπο με τον οποίο εναλλάσσεται η κατάστασή του. Παρατηρώντας το διάγραμμα αμέσως βλέπουμε ότι η τάση εναλλάσσεται με σταθερή συχνότητα. Αυτή η συχνότητα ονομάζεται συχνότητα του PWM και αντιστοιχεί σε μια χρονική περίοδο που τη συμβολίζουμε Tp. Όπως πολύ σωστά φαντάζεστε τώρα, αυτή η χρονική περίοδος ονομάζεται και περίοδος του PWM. Παρατηρούμε, επίσης, ότι το χρονικό διάστημα κατά το οποίο η τάση βρίσκεται στα 5V (Ton), είναι διαφορετικό από το χρονικό διάστημα κατά το οποίο η τάση είναι 0V (Toff). Άραγε, πώς επιδρούν αυτά τα μεγέθη στο μέσο όρο της τάσης;

Μέση τιμή

Ιδανικά, θα μας ενδιέφερε να υπολογίσουμε το μέσο όρο της τάσης σε ένα μεγάλο χρονικό διάστημα, ώστε οι τυχαίες (και μη) διακυμάνσεις να "ομαλοποιηθούν". Όμως η τάση του ακροδέκτη μεταβάλλεται περιοδικά. Επομένως, μπορούμε κάλλιστα να εστιάσουμε ακόμα και σε μία μόνο περίοδό του. Ας πάρουμε λοιπόν την περίοδο από τη χρονική στιγμή 0 ως τη στιγμή Tp (λήξη της πρώτης περιόδου). Σε αυτό το χρονικό διάστημα, ο μέσος όρος της τάσης προκύπτει αν προσθέσουμε όλες τις στιγμιαίες τιμές της και διαιρέσουμε με τη διάρκεια του χρονικού διαστήματος. Έτσι, αν ο μέσος όρος της τάσης είναι Vo, ισχύει η ακόλουθη σχέση:

$$V_o = \frac{1}{T_p}\int_0^{T_p} V(t)dt$$

Όπως βλέπετε, το άθροισμα των στιγμιαίων τιμών εκφράζεται με το ορισμένο ολοκλήρωμα της τάσης, από τη χρονική στιγμή 0 έως τη στιγμή Tp. Όμως αυτό το ολοκλήρωμα θα μπορούσε να εκφραστεί σαν το άθροισμα

δύο επιμέρους ολοκληρωμάτων. Συγκεκριμένα, η παραπάνω σχέση θα μπορούσε να γραφεί ως εξής:

$$V_o = \frac{1}{T_p}\left(\int_0^{T_{on}} V(t)dt + \int_{T_{on}}^{T_p} V(t)dt\right)$$

Τώρα, για το άθροισμα των στιγμιαίων τιμών της τάσης χρησιμοποιούμε δύο ολοκληρώματα. Το πρώτο ολοκλήρωμα δίνει το άθροισμα των τάσεων για το χρονικό διάστημα από τη στιγμή 0 έως τη στιγμή Ton. Το δεύτερο ολοκλήρωμα συνεχίζει από εκεί που σταμάτησε το πρώτο και δίνει το άθροισμα των τάσεων από τη στιγμή Ton έως τη στιγμή Tp. Όπως είναι προφανές, στο πρώτο διάστημα η τάση είναι συνεχώς 5V, ενώ στο δεύτερο συνεχώς 0V. Επομένως, ισχύουν τα ακόλουθα:

$$\int_0^{T_{on}} V(t)dt = V_s T_{on} \qquad \text{και} \qquad \int_{T_{on}}^{T_p} V(t)dt = 0T_{off}$$

Συνδυάζοντας τα παραπάνω, καταλήγουμε στην ακόλουθη σχέση:

$$V_o = \frac{T_{on}}{T_p} V_s$$

Όπως φαίνεται και στην εικόνα 1, το Tp προκύπτει από το άθροισμα των Ton και Toff. Επομένως, η τελευταία σχέση θα μπορούσε να γραφεί κι ως εξής:

$$V_o = \frac{T_{on}}{T_{on}+T_{off}} V_s$$

Σημεία προσοχής

Εξετάζοντας την τελευταία σχέση, μπορούμε να εξαγάγουμε τα χρησιμότερα συμπεράσματα για την τεχνική PWM. Όπως είπαμε νωρίτερα, το Vs αναπαριστά την τάση της λογικής κατάστασης high -- και στο Arduino είναι τα 5V. Επομένως, το Vs μπορούμε να το θεωρούμε δεδομένο και σταθερό. Ο μέσος όρος της τάσης, λοιπόν, εξαρτάται μόνο από τους χρόνους Ton και Toff. Εδώ όμως χρειάζεται λίγη προσοχή: Ο μέσος όρος της τάσης εξαρτάται από τη σχέση που έχουν μεταξύ τους τα Ton και Toff, όχι απλά από το άθροισμά τους. Με άλλα λόγια, ο μέσος όρος της τάσης *δεν εξαρτάται* από το χρονικό διάστημα Tp (το άθροισμα Ton + Toff). Όπως έχουμε πει, αυτό το χρονικό διάστημα αποτελεί την περίοδο του PWM και συνεπώς προκύπτει από τη συχνότητα του PWM. Συνοψίζοντας, λοιπόν, όταν εφαρμόζουμε την τεχνική PWM σε έναν ακροδέκτη, ο μέσος όρος της τάσης εξαρτάται μόνο από τη σχέση που έχουν μεταξύ τους τα Ton και Toff και όχι από τη συχνότητα του PWM. Παρεμπιπτόντως, η σχέση των Ton και Toff, όπως εκφράζεται στο κλάσμα αριστερά του Vs, ονομάζεται Duty Cycle (ή Duty Ratio). Το Duty Cycle θα το συναντήσετε πολύ συχνά εκφρασμένο σαν ποσοστό. Βλέπετε, η τιμή του Duty Cycle ανήκει πάντα στο κλειστό διάστημα από 0 έως 1.

Υπομονή...

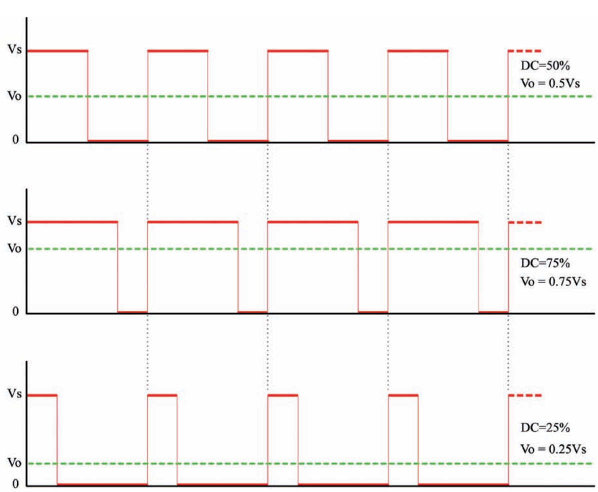
Η τεχνική που εξετάζουμε δεν είναι πολύπλοκη, αλλά δεν είναι ούτε και εύπεπτη. Για να κατανοήσετε το πώς μεταβάλλεται η μέση τάση με την τεχνική PWM, θα χρειαστεί να επανέλθετε στη θεωρία αρκετές φορές, όπως

επίσης και να πειραματιστείτε. Προφανώς, όλα αυτά θέλουν το χρόνο τους. Μέχρι τότε, θα ήταν ωφέλιμο να μελετήσετε την εικόνα 2. Σ' αυτήν εμφανίζεται η τάση ενός ακροδέκτη, όταν εφαρμόζουμε την τεχνική PWM και με διάφορες τιμές για το Duty Cycle. Ακόμα και μια γρήγορη ματιά στο διάγραμμα είναι ικανή να μας οδηγήσει σε ορισμένα συμπεράσματα, στα οποία μάς οδήγησε *και* η θεωρία. Καταρχάς, ο μέσος όρος της τάσης δεν εξαρτάται από τη συχνότητα του PWM, παρά μόνο από το Duty Cycle. Η τεχνική PWM, δηλαδή, δεν αφορά στη συχνότητα εναλλαγής της κατάστασης του ακροδέκτη. Η τεχνική επηρεάζει μόνο το χρόνο κατά τον οποίο ο ακροδέκτης βρίσκεται στη μία λογική κατάσταση, σε σχέση με το χρόνο κατά τον οποίο βρίσκεται στην άλλη λογική κατάσταση. Μιλώντας σχηματικά, η τεχνική PWM επηρεάζει μόνο το χρονικό εύρος του παλμού. Ακριβώς αυτό δηλώνει και το όνομά της: Pulse Width Modulation.

Προγραμματιστική υλοποίηση

Έχοντας ξεκαθαρίσει το τι είναι και τι δεν είναι το PWM, είμαστε σε θέση να καταστρώσουμε έναν αλγόριθμο. Μπορούμε να περιγράψουμε τη λειτουργία ενός προγράμματος, το οποίο θα εφαρμόζει την τεχνική PWM επάνω σε οποιονδήποτε ακροδέκτη, χωρίς να χρησιμοποιεί το σχετικό hardware του Arduino. Ας υποθέσουμε, λοιπόν, ότι θέλουμε να δημιουργήσουμε την τάση των 3V, επάνω σε κάποιον ακροδέκτη. Πριν ξεκινήσουμε να γράφουμε κώδικα, καλό είναι να κάνουμε κάποιους υπολογισμούς. Για την ώρα, γνωρίζουμε μόνο ότι θα πρέπει να μεταβάλουμε περιοδικά τη λογική κατάσταση του ακροδέκτη. Ωστόσο, δεν έχουμε ιδέα για το Duty Cycle και, σε τελική ανάλυση, ούτε για τους χρόνους Ton και Toff. Εδώ μπορεί να μας βοηθήσει η τελευταία σχέση, στην οποία καταλήξαμε νωρίτερα. Σε αυτήν, γνωρίζουμε τα Vs (5V) και Vo (3V). Επίσης, ξέρουμε ότι ο λόγος του χρόνου Ton προς το άθροισμα Ton + Toff, αποτελεί το Duty Cycle. Επομένως, αντικαθιστώντας τα γνωστά μεγέθη και κάνοντας τις πράξεις, υπολογίζουμε ότι το Duty Cycle είναι ίσο με 0,6. Τέλεια! Αρκεί τώρα να βρούμε δύο κατάλληλες τιμές για τα Ton και Toff. Για το Ton μπορούμε να επιλέξουμε αυθαίρετα την τιμή των 600μsec. Σε αυτή την περίπτωση, από τη σχέση που δίνει το Duty Cycle, βρίσκουμε ότι το Toff έχει την τιμή των 400μsec. Αυτό ήταν! Για να δημιουργήσουμε την τάση των 3V επάνω στον επιθυμητό ακροδέκτη, αρκεί να τον θέσουμε στη λογική κατάσταση high για 600μsec, αμέσως μετά στη λογική κατάσταση low για 400μsec, στη συνέχεια και πάλι στη λογική κατάσταση high για 600μsec και πάει λέγοντας. Τα σχετικά χρονικά διαστήματα θα μπορούσαμε να τα δημιουργήσουμε με τη συνάρτηση delayMicroseconds ή με τη βοήθεια ενός interrupt. Η πρώτη λύση είναι πολύ σπάταλη: δεσμεύει ολόκληρο το Arduino και δεν αξίζει να τη διερευνήσουμε. Ας ρίξουμε μια ματιά στη δεύτερη. Αρχικά, θα θέταμε τον ακροδέκτη στη λογική κατάσταση high και θα ρυθμίζαμε κατάλληλα έναν timer, ώστε να υπερχειλίσει μετά από 600μsec. Στη συνέχεια θα ενεργοποιούσαμε

Εικόνα 2
Η έξοδος ενός καναλιού PWM, δηλαδή η μέση τιμή της τάσης του σχετικού ακροδέκτη, καθορίζεται από το Duty Cycle και όχι από τη συχνότητα...



το σχετικό interrupt. Στη ρουτίνα εξυπηρέτησης του interrupt θα κάναμε επίσης δυο πράγματα: Θα αντιστρέφαμε τη λογική κατάσταση του ακροδέκτη και θα αλλάζαμε κατάλληλα τις ρυθμίσεις του timer. Τι εννοούμε με το κατάλληλα; Αν ο timer ήταν ρυθμισμένος ώστε το interrupt να προκύπτει μετά από 600μsec, θα τον ρυθμίζαμε ώστε το επόμενο interrupt να προκύψει στα 400μsec και αντίστροφα! Με αυτό τον τρόπο, η κατάσταση του ακροδέκτη θα άλλαζε αρχικά μετά από 600μsec, στη συνέχεια μετά από 400μsec, κατόπιν μετά από 600μsec και πάει λέγοντας...

Αυτό ήταν όλο;

Το πρόγραμμα που περιγράψαμε πετυχαίνει απόλυτα το σκοπό του: Εμφανίζει 3V, επάνω στον ακροδέκτη της επιλογής μας. Δυστυχώς, όμως, έχει αρκετές και σημαντικές ατέλειες! Καταρχάς, το να αλλάζουμε συνεχώς τις ρυθμίσεις ενός timer μέσα από τη ρουτίνα εξυπηρέτησης ενός interrupt του ίδιου του timer, μπορεί να δημιουργήσει πολλά προβλήματα. Για να κατανοήσετε το λόγο, σκεφτείτε την περίπτωση που μας ενδιαφέρει ένα πολύ μεγάλο Duty Cycle (για παράδειγμα 0,99), όπως και αυτήν που μας ενδιαφέρει ένα πολύ μικρό (για παράδειγμα 0,01). Σε αυτές τις ακραίες περιπτώσεις, το ένα από τα δύο χρονικά διαστήματα (Ton και Toff) μικραίνει εξαιρετικά πολύ σε σχέση με το άλλο. Για την ακρίβεια, μικραίνει τόσο πολύ, ώστε όταν θα έφτανε η σειρά του το αντίστοιχο interrupt θα έπρεπε να ενεργοποιηθεί *αμέσως* μετά το προηγούμενο, αν όχι λίγο νωρίτερα(!;), ώστε να προλάβουν να ολοκληρωθούν οι απαιτούμενες εργασίες στο σωστό χρόνο. Με λίγα λόγια, ενώ το πρόγραμμα που περιγράψαμε λειτουργεί άριστα για τις "ενδιάμεσες" τιμές του Duty Cycle, στις οριακές τιμές αδυνατεί να αντεπεξέλθει.

Τώρα, φανταστείτε να μας ενδιέφερε η εφαρμογή της τεχνικής PWM και σε έναν ακόμα ακροδέκτη, με διαφορετικό Duty Cycle. Εδώ τα πράγματα θα μπερδεύονταν ακόμα περισσότερο. Θα έπρεπε να επιστρατεύσουμε ένα ακόμα interrupt, και κατά πάσα πιθανότητα ενός διαφορετικού timer. Προφανώς, με αυτή τη λογική θα ξεμέναμε σύντομα από timer! Η λύση που έχουμε περιγράψει, δηλαδή, δεν είναι καθόλου scalable. Ας εστιάσουμε όμως και στην περίπτωση στην οποία θέλουμε μόνο δύο κανάλια PWM. Ανεξάρτητα από το Duty Cycle του καθενός, μετά από κάποιο χρονικό διάστημα τα δύο interrupts (ένα για κάθε κανάλι) θα τύχαινε να συμπέσουν. Ακριβώς τότε, το ένα από τα δύο κανάλια θα καθυστερούσε να ενημερωθεί. Όμως, μια καθυστέρηση στο Ton ή στο Toff, συνεπάγεται τη μεταβολή της μέσης τιμής της τάσης. Επομένως, τη στιγμή που το ένα κανάλι θα καθυστερούσε, θα μεταβαλλόταν και η τάση εξόδου του. Όπως αντιλαμβάνεστε, κάτι τέτοιο θα ήταν ανεπίτρεπτο για πάρα πολλές εφαρμογές -- χωρίς να υπολογίζουμε τα συστήματα καθοδήγησης πυραύλων. Σε ένα απλό LED, για παράδειγμα, μια στιγμιαία μεταβολή της τάσης θα γινόταν αμέσως αντιληπτή και θα ήταν μάλλον ανεπιθύμητη.

Τέλος, η λογική που περιγράψαμε για την υλοποίηση του PWM σε software παρουσιάζει ένα ακόμα πρόβλημα. Όπως θα θυμόσαστε, η τιμή για το Ton επιλέχθηκε τυχαία. Προφανώς, αυτή η τακτική για τον υπολογισμό των Ton και Toff δεν είναι και η πιο συνετή. Σκεφτείτε την περίπτωση στην οποία επιλέγουμε μια μεγάλη τιμή για το Ton και υπολογίζουμε το Toff ως προς αυτήν. Φυσικά, όσα έχουμε πει θα συνέχιζαν να ισχύουν κανονικά. Ο μέσος όρος της τάσης του ακροδέκτη δεν θα επηρεαζόταν από τη συχνότητα του PWM, αλλά μόνο από τη σχέση μεταξύ των Ton και Toff. Ωστόσο, όσο μεγαλώνουν αυτοί οι δύο χρόνοι, τόσο αυξάνεται και η πιθανότητα να αντιληφθούμε τις διαδοχικές μεταβολές στην κατάσταση

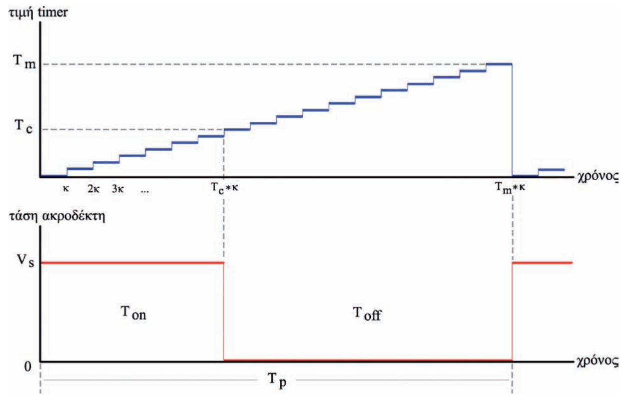
του ακροδέκτη. Με λίγα λόγια, ενώ σύμφωνα με τη θεωρία η συχνότητα του PWM μοιάζει να μην παίζει κανένα ρόλο, στην πράξη παίζει[3]. Για παράδειγμα, όταν θέλουμε να ρυθμίσουμε την ένταση ενός LED, η συχνότητα του PWM πρέπει να είναι τουλάχιστον μερικές εκατοντάδες Hz, ώστε να μη γίνεται αντιληπτή από το μάτι μας. Αντίστοιχα, όταν χρησιμοποιούμε το PWM για να συνθέσουμε ηχητικά σήματα, η συχνότητά του πρέπει να είναι μεγαλύτερη από τη μέγιστη συχνότητα που "πιάνει" το αφτί μας. Σε αυτή την περίπτωση, η συχνότητα του PWM θα πρέπει να υπερβαίνει τα 22KHz! Σε γενικές γραμμές, οι μεγάλες συχνότητες για το PWM δε βλάπτουν ποτέ. Ωστόσο, όσο αυξάνει η συχνότητα του PWM τόσο μικραίνουν οι χρόνοι Ton και Toff, με αποτέλεσμα να στενεύουν και τα χρονικά περιθώρια για τον κώδικά μας. Επομένως, όταν υλοποιούμε ένα σύστημα PWM σε software πρέπει να επιλέγουμε τη συχνότητά του με σύνεση και χωρίς υπερβολές.

Μάθημα από το hardware

Τελικά, αν και λειτουργική, η πρώτη μας υλοποίηση παρουσιάζει πολλά προβλήματα. Όλα δείχνουν ότι χρειαζόμαστε περισσότερη μελέτη. Πέρα από τη θεωρητική εξοικείωση με το PWM, θα ήταν ωφέλιμη και η μελέτη κάποιας υλοποίησης. Μη μας παρεξηγείτε. Δεν έχουμε κατά νου την αντιγραφή, παρά μόνο τη μελέτη. Αναρωτιέστε τώρα που θα βρούμε μια καλή υλοποίηση του PWM, η οποία ν' αξίζει τη μελέτη μας; Μα, στο hardware του Arduino, φυσικά! Ας εξετάσουμε, λοιπόν, τη λειτουργία ενός καναλιού PWM του Arduino...

Όπως έχουμε πει, τα κανάλια PWM του Arduino αποτελούν υποσυστήματα των timer. Τι ακριβώς συμβαίνει, λοιπόν, όταν ενεργοποιούμε κάποιο κανάλι; Καταρχάς, η λογική κατάσταση του σχετικού ακροδέκτη τίθεται σε high, ενώ την ίδια στιγμή ξεκινά ο timer στον οποίο στηρίζεται το επιλεγμένο κανάλι PWM. Τώρα, κάθε φορά που αυξάνεται η τιμή του timer το hardware ελέγχει αν αυτή η τιμή έχει φτάσει σε κάποια συγκεκριμένη στάθμη. Όταν συμβεί αυτό, ο ακροδέκτης τίθεται αυτόματα στη λογική κατάσταση low και ο timer συνεχίζει να μετράει έως ότου υπερχειλίσει. Τη στιγμή της υπερχείλισης, ο ακροδέκτης τίθεται εκ νέου στην κατάσταση high και ο κύκλος που περιγράψαμε επαναλαμβάνεται. Ωραία, θα σκεφτεί κανείς, με αυτό το σύστημα η κατάσταση του ακροδέκτη εναλλάσσεται περιοδικά, αλλά ποιος καθορίζει το Duty Cycle; Θυμηθείτε λίγο την περιγραφή. Η κατάσταση του ακροδέκτη τίθεται σε low όταν η τιμή του μετρητή φτάσει σε "κάποια συγκεκριμένη στάθμη". Επομένως, αυτή η στάθμη καθορίζει το διάστημα για το οποίο ο ακροδέκτης παραμένει στην κατάσταση high. Με άλλα λόγια, αυτή η στάθμη προσδιορίζει το χρόνο Ton. Όσο για το χρόνο Toff, αυτός ορίζεται σαν το διάστημα που απομένει από την ολοκλήρωση του Ton ως την υπερχείλιση του timer. Στην εικόνα 3 μπορείτε να δείτε τη γραφική αναπαράσταση όλων των παραπάνω. Στο διάγραμμα, το Tm δείχνει τη μέγιστη τιμή του μετρητή. Το δε Tc αντιστοιχεί στην ειδική στάθμη, η οποία συγκρίνεται κάθε στιγμή με την τιμή του μετρητή. Εδώ χρειάζεται λίγη προσοχή. Το Duty Cycle προκύπτει αν διαιρέσουμε το χρόνο που χρειάστηκε ο timer για να φτάσει στην ειδική στάθμη (χρόνος Ton), προς το χρόνο που χρειάζεται ο timer για να υπερ-

3. Αυτή η ασυμφωνία μεταξύ θεωρίας και πράξης δεν είναι νομοτελειακή. Προκύπτει από τον τρόπο με τον οποίο υπολογίσαμε το μέσο όρο της τάσης. Όπως θα θυμόσαστε, για να βρούμε τη μέση τιμή της τάσης εξετάσαμε ένα χρονικό διάστημα ίσο με την περίοδο του PWM. Όταν αυτό το διάστημα είναι πολύ μικρό, τότε αυτό που αντιλαμβανόμαστε με τις αισθήσεις μας ως μέσο όρο προσεγγίζει το μαθηματικό αποτέλεσμα. Αντίθετα, όταν η περίοδος του PWM είναι μεγάλη οι αισθήσεις μας δεν προσλαμβάνουν κανένα μέσο όρο και εντοπίζουν τις διαδοχικές μεταβολές...



Εικόνα 3
O timer αυξάνει την τιμή του και σε κάθε βήμα τη συγκρίνει με μια προκαθορισμένη στάθμη. Τη στιγμή που εξισώνονται τα δύο μεγέθη, η κατάσταση του ακροδέκτη γίνεται low. Με άλλα λόγια, τελειώνει ο χρόνος Ton και ξεκινά ο χρόνος Toff. Η περίοδος του PWM ολοκληρώνεται όταν υπερχειλίσει ο timer. Ακριβώς τότε ξεκινά το διάστημα Ton, της επόμενης περιόδου.

χειλίσει (χρόνος Ton+Toff). Βέβαια ο timer μετράει με σταθερό ρυθμό και για την αύξηση της τιμής του μεσολαβεί κάθε φορά ένα σταθερό διάστημα k. Έτσι, ο χρόνος Ton ισούται με την ειδική στάθμη επί k, ενώ ο χρόνος Ton+Toff είναι ίσος με τη μέγιστη τιμή του μετρητή επί k. Τελικά, το Duty Cycle προκύπτει αν διαιρέσουμε αυτή την περίφημη ειδική στάθμη με τη μέγιστη τιμή του timer.

Τώρα που λύσαμε κι αυτή την απορία, μπορούμε να περάσουμε σε μια ενδιαφέρουσα λεπτομέρεια. Η ειδική στάθμη στην οποία έχουμε αναφερθεί περί τις εκατό φορές ως τώρα, βρίσκεται αποθηκευμένη σε έναν καταχωρητή ο οποίος υπάγεται στο μηχανισμό του timer. Σημειώστε ότι κάθε timer διαθέτει δύο τέτοιους καταχωρητές[4]. Έτσι, με κάθε timer μπορούμε να ορίσουμε δύο διαφορετικούς χρόνους Ton και κατ' επέκταση δύο διαφορετικά Duty Cycle. Ακριβώς γι' αυτό, κάθε timer του Arduino προσφέρει και δύο κανάλια PWM. Άραγε, τι θα γινόταν αν κάποιος timer διέθετε πέντε τέτοιους καταχωρητές; Σε αυτή την περίπτωση, για τον συγκεκριμένο timer θα μπορούσαμε να προσδιορίσουμε πέντε διαφορετικές τιμές για το Ton. Συνεπώς, ο συγκεκριμένος timer θα μπορούσε να μας προσφέρει πέντε κανάλια PWM! Ενδιαφέρουσα σκέψη ;)

Επιστροφή στο software

Όπως είδαμε, το hardware του Arduino αξιοποιεί πολύ έξυπνα τους timer για την υλοποίηση της τεχνικής PWM. Για την ακρίβεια, τους αξιοποιεί τόσο έξυπνα που μας βάζει σε σκέψεις: Μήπως θα μπορούσαμε να υλοποιήσουμε έναν μετρητή στο software; Κάνοντας κάτι τέτοιο, θα μπορούσαμε να στήσουμε γύρω του έναν μηχανισμό παρόμοιο με εκείνον του hardware και να κατασκευάσουμε τελικά ένα σύστημα PWM. Παίζοντας με το software, θα είμαστε σε θέση να ορίσουμε όσες ειδικές στάθμες θέλουμε και, κατ' επέκταση, όσους χρόνους Ton θέλουμε. Έτσι, γύρω από τον ίδιο μετρητή θα ορίζαμε πολλά διαφορετικά Duty Cycle και αντίστοιχου πλήθους κανάλια PWM. Όπως αντιλαμβάνεστε, η δημιουργία ενός μετρητή στο software αποτελεί πανεύκολη υπόθεση. Άρα, η συγκρότηση ενός πανίσχυρου μηχανισμού PWM είναι εφικτή και ευκολότερη απ' όσο νομίζαμε αρχικά. Ας περιγράψουμε λοιπόν ένα πρόγραμμα, το οποίο υλοποιεί ένα σύστημα PWM των δέκα καναλιών...

Το σημαντικότερο "εξάρτημα" σε αυτό το πρόγραμμα αποτελεί ο μετρητής. Η τιμή του θα πρέπει να αυξάνεται με ακρίβεια, ενώ η λειτουργία του δεν θα πρέπει να εμποδίζει τις υπόλοιπες εργασίες του προγράμματος. Όπως καταλαβαίνετε, αυτές οι απαιτήσεις παραπέμπουν αυτόματα σε interrupt. Πριν όμως αρχίσουμε να παίζουμε με κάποιον hardware timer και τα διαθέσιμα interrupts, πρέπει να αποφασίσουμε κάθε πότε θα αυξάνεται η τιμή του μετρητή μας, καθώς και ποια θα είναι η μέγιστη τιμή του. Βλέπετε, η μέγιστη τιμή του μετρητή θα καθορίζει το πλήθος των

4. Τα ονόματα των εν λόγω καταχωρητών είναι OCRxA και OCRxB, όπου x είναι ο χαρακτηριστικός αριθμός του εκάστοτε timer. Αυτά τα ομολογουμένως παράξενα ονόματα, προκύπτουν από τις λέξεις Output Compare Register.

διαφορετικών τιμών, τις οποίες θα μπορούμε να επιλέξουμε για το χρόνο Ton. Ένας μετρητής που τερματίζει στην τιμή 3, επιτρέπει να ορίσουμε τέσσερεις μόνο τιμές για το χρόνο Ton (0, 1, 2 και 3). Επομένως, γύρω από ένα τέτοιο μετρητή μπορούμε να ορίσουμε τέσσερεις μόνο τιμές για το Duty Cycle (0%, 25%, 50% και 75% αντίστοιχα). Βέβαια, η αλήθεια είναι ότι μπορούμε πολύ εύκολα να ορίσουμε και πρόσθετη τιμή για το Duty Cycle (100%). Ωστόσο, ακόμα και με πέντε τιμές για το Duty Cycle, δεν θα έχουμε λεπτομερή έλεγχο τον εξόδων PWM. Για να ελέγχουμε με ακρίβεια τις τάσεις που θα παράγει το σύστημά μας, ο μετρητής μας θα πρέπει να τερματίζει σε αρκετά μεγαλύτερη τιμή. Θα πρέπει για παράδειγμα να τερματίζει στο 100, ώστε να έχουμε τουλάχιστον 100 υποδιαιρέσεις για το Duty Cycle. Αφού ξεμπερδέψαμε με τη μέγιστη τιμή του μετρητή, πρέπει να εξετάσουμε και το πόσο γρήγορα θα μετράει. Ο ρυθμός του μετρητή μας, σε συνδυασμό με τη μέγιστη τιμή του, καθορίζει το χρόνο Ton+Toff. Με άλλα λόγια, ο χρόνος ανά τον οποίο αυξάνεται η τιμή του μετρητή μας, σε συνδυασμό με τη μέγιστη τιμή του, καθορίζει την περίοδο του PWM. Αν υποθέσουμε ότι θέλουμε μια συχνότητα PWM κοντά στο 1KHz, η περίοδος του PWM θα είναι 1msec. Επομένως, αν ο μετρητής μας υπερχειλίζει στην τιμή 100, τότε θα πρέπει να αυξάνει την τιμή του ανά μόλις 10μsec. Μετά κι από αυτόν τον υπολογισμό, μπορούμε να επιστρέψουμε στον κώδικα. Για να πετύχουμε το σκοπό μας, θα πρέπει να ενεργοποιήσουμε ένα interrupt ενός εκ των hardware timer. Φυσικά, ο timer θα πρέπει να ρυθμιστεί ώστε το σχετικό interrupt να ενεργοποιείται ανά 10μsec. Στη ρουτίνα εξυπηρέτησης του συγκεκριμένου interrupt, θα πραγματοποιούνται τρεις εργασίες. Καταρχάς, θα αυξάνεται η τιμή του μετρητή μας. Στη συνέχεια, αν θέλουμε να υλοποιήσουμε πέντε κανάλια PWM τότε θα συγκρίνουμε την τιμή του μετρητή με τις πέντε τιμές, οι οποίες προσδιορίζουν το Duty Cycle κάθε καναλιού. Τέλος, σύμφωνα με το αποτέλεσμα των συγκρίσεων, θα εναλλάσσουμε κατάλληλα την κατάσταση των ακροδεκτών, τους οποίους ελέγχουν τα κανάλια PWM. Αυτό ήταν όλο!

Το κύκλωμα

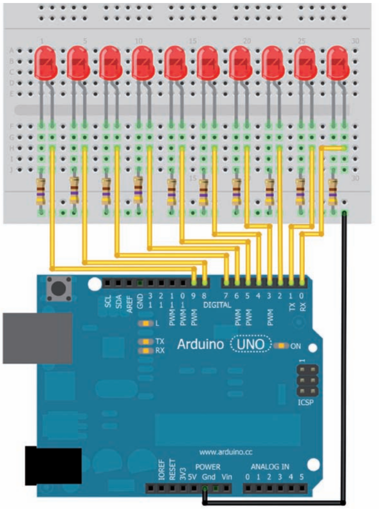
Επιτέλους, μετά από αρκετή θεωρία, είμαστε έτοιμοι να ξεκινήσουμε το παιχνίδι με την κατασκευή που υποσχεθήκαμε. Για το larson scanner θα χρειαστούμε λίγα και απλά εξαρτήματα. Συγκεκριμένα, θα χρειαστούμε δέκα LED και δέκα αντιστάσεις προστασίας. Τα LED και οι αντιστάσεις θα συνδεθούν ανάμεσα στους πρώτους δέκα ακροδέκτες του Arduino και στη γείωση (GND). Οι άνοδοι των LED θα βρίσκονται από την πλευρά των ακροδεκτών του Arduino, ώστε να ενεργοποιούνται κάθε φορά που ο αντίστοιχος ακροδέκτης βρίσκεται στην κατάσταση high. Τέλος, οι αντιστάσεις προστασίας πρέπει να είναι μεγαλύτερες από 470Ω. Αυτός ο περιορισμός έχει να κάνει με τη σύνδεση ενός LED στον ακροδέκτη 0. Ο συγκεκριμένος ακροδέκτης συνδέεται στη σειριακή είσοδο του Arduino κι αν δεν είναι καλά "απομονωμένος" από τη γείωση η μεταφορά του κώδικα στο Arduino θα αποτυγχάνει. Θυμηθείτε, λοιπόν, να χρησιμοποιήσετε αντιστάσεις των 470Ω ή μεγαλύτερες. Το κύκλωμά μας μπορείτε να το δείτε στις εικόνες 4 και 5.

Ο κώδικας

Πριν ξεκινήσουμε τη μελέτη του κώδικα, κατεβάστε τον από το ακόλουθο URL:

```
http://bit.ly/dh007pwmsrc
```

Στο πακέτο που θα πάρετε υπάρχουν τέσσερα διαφορετικά προγράμματα. Τα δύο από αυτά πετυχαίνουν αυτό που σας έχουμε υποσχεθεί: Σχηματίζουν ένα larson scanner σαν αυτόν του KITT και έναν ακόμα, σαν το μάτι των Cylon. Για την υλοποίηση αυτών των εφέ τα δύο προγράμματα ενσωματώνουν ένα μηχανισμό PWM, με τη βοήθεια του οποίου αναβοσβήνουν ομαλά (fade in/fade out) τα LED. Τον ιδιο μηχανισμό PWM ενσωματώνουν και τα υπόλοιπα δύο προγράμματα. Ωστόσο πρόκειται για απλούστερες εφαρμογές που δεν πετυχαίνουν κάποιο σύνθετο εφέ, αλλά επιδεικνύουν την αναλυτικότητα (την ακρίβεια) που προσφέρει ο μηχανισμός μας. Στη συνέχεια θα εξετάσουμε τον κώδικα που σχηματίζει το δικό μας σύστημα PWM...



Εικόνα 4
Αυτή τη φορά το κύκλωμά μας είναι πολύ απλό. Τα LED πρέπει να συνδεθούν με την άνοδο προς τους ακροδέκτες του Arduino. Έτσι, όταν κάποιος ακροδέκτης μεταβαίνει στην κατάσταση high, το αντίστοιχο LED θα ανάβει. Προσοχή χρειάζεται και στις αντιστάσεις προστασίας, οι οποίες θα πρέπει να έχουν τιμή 470Ω ή μεγαλύτερη. Σε διαφορετική περίπτωση η σειριακή θύρα του συστήματος (ο ακροδέκτης 0) δεν θα λειτουργεί σωστά και θα είναι αδύνατο να προγραμματίσουμε το Arduino.

```
cli();
TCCR2A = 0; // clear TCCR2A register
TCCR2B = 0; // clear TCCR2B register
TCCR2A |= (1 << WGM21); // select CTC mode
TCCR2B |= (1 << CS21); // set prescaler to 256
TCCR2B |= (1 << CS22); //
OCR2A = 0; // timer2 top counting value
TIMSK2 |= (1 << OCIE2A); // enable timer compare interrupt
sei();
```

Το παραπάνω απόσπασμα προέρχεται από τη συνάρτηση setup και είναι πανομοιότυπο σε όλα μας τα προγράμματα. Το συγκεκριμένο τμήμα ρυθμίζει τον timer2 και ενεργοποιεί ένα από τα interrupt του εν λόγω timer. Συγκεκριμένα, επιλέγει τον τρόπο λειτουργίας Clear on Timer Compare match (CTC), ο οποίος επιβάλει στον εκάστοτε timer να μηδενίζεται κάθε φορά που φτάνει σε μια προκαθορισμένη τιμή. Για τον timer2, αυτή η τιμή προσδιορίζεται από τον καταχωρητή OCR2A. Στο πρόγραμμά μας, στο συγκεκριμένο καταχωρητή δίνουμε την τιμή 0. Έτσι, ο timer2 θα μηδενίζεται και αμέσως μετά θα υπερχειλίζει σε κάθε χτύπο του ρολογιού του. Παρεμπιπτόντως, το ρολόι του timer2 προκύπτει από το κεντρικό ρολόι του επεξεργαστή του Arduino, αφού διαιρεθεί με το 256 (prescaler = 256). Στη συνέχεια, ο κώδικας ενεργοποιεί το interrupt, το οποίο προκύπτει όταν ο timer2 φτάνει στην τιμή OCR2A. Σύμφωνα με τις ρυθμίσεις που έχουν προηγηθεί, το συγκεκριμένο interrupt θα πραγματοποιείται σε κάθε χτύπο του ρολογιού του timer2 και συνεπώς με συχνότητα 62.5KHz[5].

Μέχρι στιγμής το πρόγραμμά μας έχει ενεργοποιήσει ένα interrupt, το οποίο θα εκτελείται με σχετικά μεγάλη συχνότητα. Ας δούμε τώρα τι συμβαίνει στη ρουτίνα εξυπηρέτησης αυτού του interrupt.

5. Όπως είπαμε, η συχνότητα του ρολογιού του timer2 προκύπτει από τη συχνότητα του επεξεργαστή του Arduino, αφού διαιρεθεί με το 256. Επομένως, ισούται με 16MHz/256 = 62.5KHz.

```
ISR(TIMER2_COMPA_vect)
{
  PORTD = portd_data;
  PORTB = portb_data;

  pwm_cnt++;
  pwm_cnt = pwm_cnt & 15;

  if (pwm_cnt == 0) {counter++;}

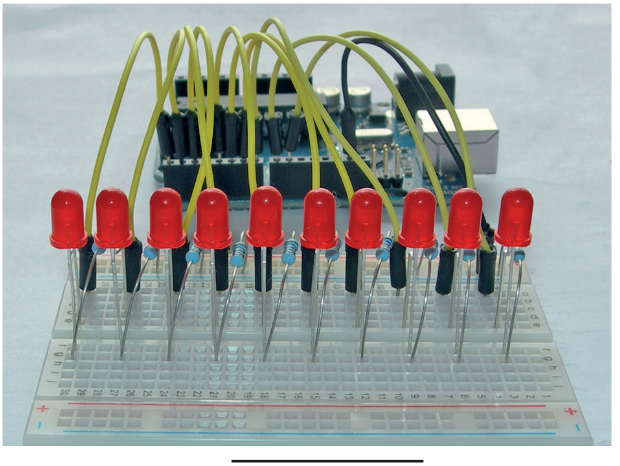
  portd_data = 0;
  portb_data = 0;

  if (levels[0] > pwm_cnt) {portd_data |= 1;}
  if (levels[1] > pwm_cnt) {portd_data |= 2;}
  if (levels[2] > pwm_cnt) {portd_data |= 4;}
  if (levels[3] > pwm_cnt) {portd_data |= 8;}
  if (levels[4] > pwm_cnt) {portd_data |= 16;}
  if (levels[5] > pwm_cnt) {portd_data |= 32;}
  if (levels[6] > pwm_cnt) {portd_data |= 64;}
  if (levels[7] > pwm_cnt) {portd_data |= 128;}
  if (levels[8] > pwm_cnt) {portb_data |= 1;}
  if (levels[9] > pwm_cnt) {portb_data |= 2;}
}
```

Για την ώρα μπορείτε να αγνοήσετε τις πρώτες δύο γραμμές. Η παρουσία τους στη συγκεκριμένη θέση θα εξηγηθεί αργότερα. Στις αμέσως επόμενες γραμμές, η τιμή της μεταβλητής pwm_cnt αυξάνεται. Πρόκειται για έναν μετρητή γύρω από τον οποίο έχουμε στήσει το σύστημα PWM. Όπως βλέπετε, αφού αυξήσουμε την τιμή του εν λόγω μετρητή, πραγματοποιούμε τη λογική πράξη AND με τον αριθμό 15 (ή `00001111` στο δυαδικό, βλ. και http://deltahacker.gr/?p=4192). Με αυτό το τέχνασμα εξασφαλίζουμε ότι ο μετρητής μας θα φτάνει πάντα ως την τιμή 15 και στη συνέχεια θα μηδενίζεται! Ως εκ τούτου, το σύστημα PWM που κατασκευάζουμε θα μας επιτρέπει να ορίζουμε 15 διαφορετικές τιμές για το Duty Cycle[6]. Αν επιθυμούσαμε περισσότερες επιλογές, για μεγαλύτερη ακρίβεια στην έξοδο του PWM, θα μπορούσαμε να θέσουμε ένα μεγαλύτερο όριο για το μετρητή μας. Για παράδειγμα, θα θέταμε ως όριο το 100, αντικαθιστώντας τις σχετικές γραμμές κώδικα με τις ακόλουθες:

```
pwm_cnt++;
if (pwm_cnt == 101) { pwm_cnt = 0;}
```

Μετά από την αύξηση της τιμής του



Εικόνα 5
Η εντυπωσιακή συστοιχία LED περιμένει τον κώδικά μας. Εμείς πάλι περιμένουμε να δούμε ζωντανά τα εφέ που θα σχεδιάσουμε και θα υλοποιήσουμε στον κώδικα!

6. Στην πράξη, αν συνυπολογίσουμε και την οριακή περίπτωση στην οποία το Duty Cycle είναι ίσο με τη μονάδα, θα μπορούμε να ορίσουμε δεκαέξι διαφορετικές τιμές...

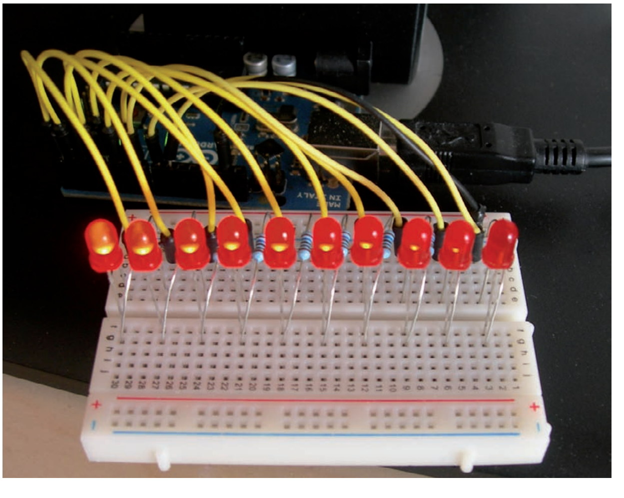
μετρητή, ελέγχουμε αν έχει ολοκληρωθεί μια περίοδος του PWM. Αυτό συμβαίνει κάθε φορά που ο μετρητής μας υπερχειλίζει κι επιστρέφει στην τιμή 0. Κάθε φορά που συμβαίνει αυτό, αυξάνουμε την τιμή ενός ακόμα μετρητή. Ο λόγος γίνεται για τη μεταβλητή counter, η οποία αξιοποιείται σε άλλα τμήματα του προγράμματος για τη μέτρηση του χρόνου. Για παράδειγμα, στο πρόγραμμα που υλοποιεί το φωτεινό σηματάκι του KITT, αυτή η μεταβλητή χρησιμοποιείται ως ένα χρονόμετρο που σηματοδοτεί τη μετατόπιση της φωτεινής κουκίδας. Όλα αυτά όμως ξεφεύγουν από το θέμα μας. Ας επιστρέψουμε στην υλοποίηση του PWM. Αφού ενημερωθούν κατάλληλα οι δύο μετρητές, το πρόγραμμα καθαρίζει τις μεταβλητές portb_data και portd_data. Σε αυτές τις μεταβλητές αποθηκεύεται η *μελλοντική* κατάσταση των ακροδεκτών, οι οποίοι ελέγχονται από τα κανάλια PWM. Λέγοντας μελλοντική, αναφερόμαστε σ' εκείνη την κατάσταση στην οποία θα πρέπει να βρεθούν οι ακροδέκτες στον επόμενο χτύπο του μετρητή μας (την επόμενη φορά που θα εκτελεστεί η ρουτίνα εξυπηρέτησης του interrupt). Αμέσως μετά, το πρόγραμμα συγκρίνει την τιμή του μετρητή με τα περιεχόμενα ενός πίνακα. Σε αυτόν τον πίνακα έχουμε αποθηκεύσει τις τιμές που καθορίζουν το Duty Cycle κάθε καναλιού. Σύμφωνα με το αποτέλεσμα κάθε σύγκρισης, το πρόγραμμα ενημερώνει το αντίστοιχο bit στις μεταβλητές portd_data και portb_data. Όπως αντιλαμβάνεστε, οι τιμές αυτών των μεταβλητών θα πρέπει κάποια στιγμή να αντιγραφούν στους Port Register του συστήματος, ώστε να ενημερωθούν οι ακροδέκτες. Η καταλληλότερη στιγμή για να γίνει αυτό είναι η αρχή του ISR. Ακριβώς γι' αυτό, η ρουτίνα εξυπηρέτησης του interrupt ξεκινά με τις εξής δύο γραμμές:

```
PORTD = portd_data;
PORTB = portb_data;
```

Σε ένα αξιοπρεπές σύστημα PWM οι ακροδέκτες πρέπει να ενημερώνονται ταυτόχρονα -- και ταχύτατα. Για να πετύχουμε τον πρώτο στόχο, οι ακροδέκτες ενημερώνονται μαζικά με τη βοήθεια των Port Register. Για να πετύχουμε το δεύτερο στόχο, η ενημέρωση των Port Register πραγματοποιείται πριν από οποιαδήποτε άλλη εργασία, εφόσον κάποιες από αυτές έχουν μη προβλέψιμο χρόνο εκτέλεσης. Αναφερόμαστε στις δομές if, το περιεχόμενο των οποίων δεν εκτελείται πάντα.

Βουρ στο παιχνίδι!

Μετά από όλα αυτά ελπίζουμε να φωτίσαμε κάθε πτυχή των συστημάτων PWM και να κεντρίσαμε το ενδιαφέρον σας. Η τεχνική PWM βρίσκει πάρα πολλές εφαρμογές, από τη ρύθμιση της φωτεινότητας ενός απλού LED ως τη ρύθμιση της ταχύτητας ενός μοτέρ, σε ένα ρομποτικό μηχανισμό ακριβείας. Ελπίζουμε να σας φανούν ενδιαφέροντα τα προγραμματάκια μας αλλά, πολύ περισσότερο, ελπίζουμε να δούμε σύντομα τις δικές σας υλοποιήσεις.



Εικόνα 6

Το σύστημα PWM των δέκα καναλιών είναι έτοιμο! Στη φωτογραφία φαίνεται μία από τις πρώτες δοκιμές μας, πριν υλοποιήσουμε τους δύο larson scanner. Το κανάλι PWM που οδηγεί το άκρα αριστερό LED, έχει Duty Cycle ίσο με τη μονάδα (100%). Τα υπόλοιπα LED οδηγούνται από κανάλια PWM, τα οποία παρουσιάζουν συνεχώς μειούμενο Duty Cycle.

Skill: Beginner

Tags: Python, pygame, objects, event, object oriented programming, event driven programming, frames, framerate

Python programming: (Part 3) Pygame powered games!

by Sonic

Στο παρόν άρθρο θα εξερευνήσουμε τις δυνατότητες του Pygame. Πρόκειται για ένα καταπληκτικό module της Python, το οποίο μάς επιτρέπει να φτιάξουμε παιχνίδια γραφικών σχετικά εύκολα. Η γνωριμία θα ξεκινήσει με την εγκατάσταση του Pygame, ενώ στη συνέχεια θα μάθουμε πώς λειτουργούν τα προγράμματα που τρέχουν σε παράθυρα. Αυτό θα γίνει με τη βοήθεια μερικών απλών παραδειγμάτων. Πάντως μη νομίζετε ότι οι εφαρμογές μας θα είναι περιορισμένες στα Windows. Τα καμάρια μας θα τρέχουν σε κάθε λειτουργικό με παραθυρικό περιβάλλον!

Μέχρι τώρα σας έχουμε δείξει απλά παιχνίδια, τα οποία θα μπορούσατε κάλλιστα να παίζατε σ' ένα τερματικό του PDP11! Πρέπει βέβαια να ομολογήσουμε ότι ένα adventure κειμένου δεν θα είχε καλή τύχη στις μέρες μας. Αυτή η εισαγωγή όμως ήταν απαραίτητη για δύο λόγους: Για να μάθετε τα βασικά της Python και για ν' αποκτήσετε αλγοριθμικό τρόπο σκέψης. Πιστεύουμε ότι με το text adventure πετύχαμε και τα δύο!

Είναι προφανές ότι η γνωριμία σας με την Python δεν θα ήταν σωστό να ξεκινήσει μ' εφαρμογές γραφικών. Σ' αυτή την περίπτωση θα έπρεπε να μάθετε πάρα πολλά πράγματα, σ' ελάχιστο χρόνο. Αυτό όχι μόνο δεν είναι εύκολο, αλλά δεν είναι και καθόλου διασκεδαστικό. Οι εφαρμογές γραφικών σχεδιάζονται και λειτουργούν πολύ διαφορετικά, σε σχέση με τις εφαρμογές της κονσόλας. Αν είχαμε ξεκινήσει με τέτοια προγράμματα, θα έπρεπε να κατανοήσετε όλες αυτές τις ιδιαιτερότητες κι επιπλέον να μάθετε όλα τα δομικά στοιχεία της Python, ταυτόχρονα. Τώρα, βέβαια, πιστεύουμε αυτή η δυσκολία έχει ξεπεραστεί: Έχετε μάθει τα βασικά της γλώσσας[1] κι έχετε φτιάξει μερικά απλά παιχνιδάκια για τη γραμμή εντολών. Επομένως, νομίζουμε ότι μπορείτε να περάσετε στις πολύ πιο ενδιαφέρουσες εφαρμογές γραφικών. Αν δεν το 'χετε συνειδητοποιήσει, σ' αυτές ανήκουν κι όλα τα σύγχρονα παιχνίδια! Έτσι, για να διευκολύνουμε αυτή τη μετάβαση, θ' ασχοληθούμε με το module pygame. Νομίζουμε ότι τ' όνομα του μιλάει από μόνο του ;)

Εγκατάσταση του pygame

Μην περιμένετε τίποτα πολύπλοκο ή δύσκολο. Αρχικά, πρέπει να επισκεφτείτε τη δικτυακή του τοποθεσία www.pygame.org. Αν χρησιμοποιείτε Windows, μπορείτε να κατεβάσετε την τελευταία σταθερή έκδοση του pygame, για την Python 2.7 (αυτήν έχουμε εγκαταστήσει και μ' αυτήν παίζουμε). Τη στιγμή που γράφεται το άρθρο, η καλύτερη επιλογή για κατέβασμα είναι η εξής:

```
http://pygame.org/ftp/pygame-1.9.2a0.win32-py2.7.msi
```

Αν χρησιμοποιείτε 64bit Windows και την 64bit Python, κατεβάστε την αντίστοιχη (δοκιμαστική) έκδοση του pygame από εδώ:

```
http://www.lfd.uci.edu/~gohlke/pythonlibs/#pygame
```

Στην περίπτωση που χρησιμοποιείτε κάποια διανομή Linux βασισμένη στο Debian, μπορείτε να εγκαταστήσετε το pygame δίνοντας κάτι τέτοιο:

```
# apt-get install python-pygame
```

Στο FreeBSD η εγκατάσταση γίνεται εξίσου εύκολα:

```
# cd /usr/ports/devel/py-game
# make install clean
```

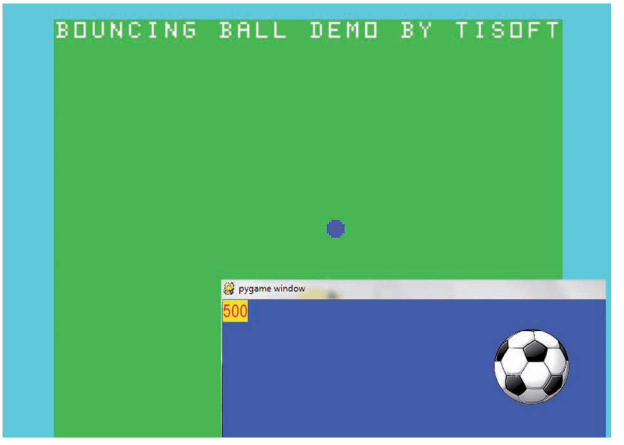
Τέλος, για τους κρυφούς χρήστες του OS X (το ξέρουμε ότι ζείτε ανάμεσά μας), το Lion διαθέτει ήδη την python 2.7 από τη "μαμά" του. Επομένως θα χρειαστείτε μόνο το pygame, το οποίο θα βρείτε εδώ:

```
www.pygame.org/ftp/pygame-1.9.2pre-py2.7-macosx10.7.mpkg.zip
```

Αν πάντως τρέχετε κάποιο άλλο λειτουργικό και δεν σας καλύψαμε, μπορείτε ν' ανατρέξετε στη σελίδα `www.pygame.org/download.shtml`. Στην περίπτωση των Windows και του OS X, αφού κατεβάσετε το σχετικό πα-

1. Η παρουσίαση των βασικών πτυχών της Python δεν έχει τελειώσει. Η γνωριμία με τα βασικά της γλώσσας θα συνεχιστεί στα άρθρα με αριθμό της μορφής x.5, στο site του περιοδικού.

κέτο θα πρέπει να το εκτελέσετε για να εγκατασταθεί. Αυτό ήταν όλο! Τώρα μπορούμε να δούμε μια όμορφη υλοποίηση του Hello World, σε pygame!



Το παλιό και το καινούριο: Μια εικόνα του πρωτότυπου bouncing ball, όπως το έγραψα το 1985 στον TI-99/4A. Είναι εμφανώς... retro, αλλά φυσικά έχει πλάκα! Την εποχή εκείνη η κίνηση δεν γινόταν με καρέ (όπου συνήθως ξανασχεδιάζουμε όλη την οθόνη) αλλά με sprites ή κίνηση χαρακτήρων, όπου απλά μετακινούμε το αντικείμενο στη νέα του θέση και το σβήνουμε από την παλιά. Δεν θα ήταν άλλωστε δυνατόν για ένα μηχάνημα της εποχής να ανασχεδιάζει μια ολόκληρη οθόνη 50 με 60 φορές το δευτερόλεπτο. Φυσικά, ούτε λόγος για double buffering!

Hello Pygame!

Επειδή δεν θέλουμε να σας κρατάμε σε αγωνία, ορίστε το πρόγραμμά μας:

http://bit.ly/hellopygame

Αφού το κατεβάσετε, ανοίξτε το με τον αγαπημένο σας text editor. Όσοι αποφύγατε το σοκ από το μέγεθος του προγράμματος, μπορείτε να συνεχίσετε. Θα σας αποδείξουμε ότι δεν είναι τόσο περίπλοκο όσο φαίνεται και ότι ήταν καλή ιδέα να μάθετε προγραμματισμό σε Python με το pygame. Για όσους δεν απέφυγαν το σοκ, μην ανησυχείτε καθόλου: Πρόκειται για μια απλή κρίση υπογλυκαιμίας. Φάτε ένα σοκολατάκι και προχωράμε :)

Θα μελετήσουμε το πρόγραμμα γραμμή προς γραμμή και ταυτόχρονα θα εξετάσουμε όλες τις νέες έννοιες. Θα αναφερθούμε στον αντικειμενοστρεφή προγραμματισμό (object oriented programming), στα συμβάντα (events) και στον προγραμματισμό που "οδηγείται" από συμβάντα ή αλλιώς event driven programming. (Σ.τ.Ε. Κάποτε είχα διαβάσει την εξής απόδοση: "γεγονοδηγούμενος προγραμματισμός". Αυτό.)

```
import pygame
from pygame.locals import *
from sys import exit
```

Με την πρώτη εντολή δηλώνουμε τη χρήση της βιβλιοθήκης pygame. Όπως φαντάζεστε, οτιδήποτε αξιοποιούμε από το pygame θα ξεχωρίζει αμέσως, καθώς θα έχει τη μορφή "pygame.εντολή". Η δεύτερη γραμμή κώδικα απαιτεί αναλυτική εξήγηση: Το pygame.locals περιέχει ένα σύνολο από σταθερές, οι οποίες είναι ιδιαίτερα χρήσιμες όταν προγραμματίζουμε με το pygame. Για παράδειγμα αντιστοιχεί ονόματα στους κωδικούς των πλήκτρων, δηλαδή στους αριθμούς που επιστρέφει το πληκτρολόγιο καθώς χτυπάμε τα πλήκτρα σε μια απέλπιδα προσπάθεια να γλυτώσουμε το διαστημόπλοιο από τα εχθρικά πυρά.

Η λέξη from, σε συνδυασμό με την import, επιτρέπει να εισαγάγουμε σ' ένα πρόγραμμα τα στοιχεία ακριβώς που θέλουμε, από ένα αρχείο ή module. Στα δεξιά της γραμμής έχουμε βάλει ένα αστεράκι κι αυτό σημαίνει ότι θέλουμε να εισαγάγουμε στο πρόγραμμα όλα όσα περιέχει το pygame.locals. Εδώ η χρήση του αστερίσκου έχει κι ένα ακόμα αποτέλεσμα: Στο εξής, όταν θέλουμε να χρησιμοποιήσουμε μια συνάρτηση από το pygame δεν θα είμαστε υποχρεωμένοι να γράφουμε πρώτα το όνομά του μαζί με μια τελεία. Αν αναρωτιέστε γιατί δεν το κάνουμε αυτό με όλα τα module, σας αφήνουμε να το σκεφτείτε σαν άσκηση. Αν δεν το βρείτε, θα σας το πούμε στο άρθρο 3.5. Ξέρετε πού θα το βρείτε αυτό, έτσι δεν είναι; (Σ.τ.Ε. Είμαι πολύ καλούλης εγώ και δεν θέλω να παιδεύω κόσμο: http://deltahacker.gr/?p=5124.)

Στην τρίτη γραμμή από το module sys εισάγουμε τη συνάρτηση exit, η οποία απαιτείται για να μπορούμε να τερματίζουμε όμορφα ένα πρόγραμμα. Σε μερικές πλατφόρμες (ονόματα δεν λέμε, ούτε δείχνουμε τα Windows), αν δεν καλέσετε την exit() στο τέλος του προγράμματός σας θ' αντιμετωπίσετε παράξενα μηνύματα λάθους και κολλήματα.

```
windowsize = (320,140)
```

Το windowsize αποτελεί ένα tuple. Όπως θα θυμόσαστε, τα tuples μοιάζουν με λίστες αλλά έχουν ορισμένους περιορισμούς. Στο συγκεκριμένο tuple αποθηκεύουμε το μέγεθος (πλάτος και ύψος) του παραθύρου της εφαρμογής μας. Αργότερα, το windowsize θα χρησιμοποιηθεί σαν παράμετρος σε μια συνάρτηση του pygame. Σημειώστε ότι οι περισσότερες συναρτήσεις του pygame, δέχονται δεδομένα σε μορφή tuple.

Για την ώρα θα παρακάμψουμε την ερμηνεία των centerMessage και getQuit. Θα επιστρέψουμε αργότερα σ' αυτές τις συναρτήσεις, αφού κατανοήσουμε το κύριο μέρος του προγράμματος...

```
def main():
    pygame.init()
```

Η init αποτελεί την πρώτη συνάρτηση του pygame, την οποία πρέπει να καλούμε σε ένα πρόγραμμα. Όπως δηλώνει και τ' όνομά της, αρχικοποιεί τη λειτουργία του pygame.

```
surfacecolor = (50,80,250)
```

Σε αυτό το tuple ορίζουμε το χρώμα της επιφάνειάς μας, ορίζοντας τιμές για το κόκκινο, το πράσινο και το μπλε αντίστοιχα. Ο συνδυασμός που επιλέξαμε εμείς δίνει ένα ενδιαφέρον γαλαζοπράσινο χρωματάκι, το οποίο θ' αποτελεί το φόντο της εφαρμογής μας.

```
screen = pygame.display.set_mode(windowsize, DOUBLEBUF)
```

Το παραπάνω δημιουργεί την βασική μας επιφάνεια: Το παράθυρο μέσα στο οποίο θα εμφανιστούν τα πάντα. Παρατηρήστε ότι το μέγεθος το ορίζουμε με τη βοήθεια του tuple, που ορίσαμε παραπάνω. Το DOUBLEBUF (μια συμβολική σταθερά που πήραμε από το pygame.locals) μπορείτε να το αγνοήσετε για την ώρα. Ο ρόλος του θα εξηγηθεί παρακάτω.

```
pygame.display.set_caption("Hello Pygame!")
```

Ο τίτλος του παραθύρου. Ξέρετε, αυτό το ενοχλητικό κείμενο που εμφανίζεται στη μπάρα του παραθύρου και μας εκνευρίζει, την ώρα που προσπαθούμε να μετακινήσουμε το παράθυρο με το ποντίκι.

```
textfont = pygame.font.SysFont("Arial",48)
```

Εδώ δημιουργούμε ένα αντικείμενο font. Με τη βοήθεια αυτού του αντικειμένου θα μπορούμε αργότερα να εμφανίζουμε μηνύματα. Για να μιλήσουμε πιο απλά, ας πούμε ότι επιλέγουμε τη γραμματοσειρά με την οποία θα εμφανιστεί το κείμενό μας.

```
thetext = textfont.render("Hello Pygame!", True, (255,0,0),(255,255,0))
```

Εδώ δημιουργούμε την επιφάνεια thetext, η οποία περιέχει τη φράση Hello Pygame! Τα δύο tuples που βλέπετε αντιστοιχούν στο χρώμα προσκηνίου (foreground) και παρασκηνίου (background), αντίστοιχα. Με τις τιμές που έχουμε επιλέξει θα 'χουμε κόκκινα γράμματα πάνω σε κίτρινο φόντο. Το True ενεργοποιεί το antialiasing για το κείμενο. Μπορείτε κάλλιστα να δοκιμάσετε και το False, για να δείτε τι κάθεται καλύτερα στο μάτι σας.

Νέες έννοιες

Δεν ξέρουμε αν το προσέξατε, αλλά έχουμε αναφέρει τις λέξεις "επιφάνεια" και "αντικείμενο", χωρίς να τις ερμηνεύσουμε. Θα επανορθώσουμε αμέσως! Οτιδήποτε θέλουμε να εμφανίσουμε στο pygame τοποθετείται πάνω σ' ένα κατασκεύασμα που ονομάζεται επιφάνεια. Στο πρόγραμμά μας, η πρώτη επιφάνεια ήταν το ίδιο το παράθυρο της εφαρμογής, το οποίο δημιουργήσαμε με τη μεταβλητή screen. Ακολούθως δημιουργήσαμε μια ακόμη επιφάνεια, με τη μεταβλητή thetext, για την προβολή ενός μηνύματος. Σας ακούμε να ρωτάτε, τώρα, τι είδους μεταβλητές είναι η screen και η thetext. Μέχρι στιγμής έχετε δει μεταβλητές οι οποίες περιέχουν strings, λίστες ή αριθμούς. Τι σόι μεταβλητή μπορεί να διατηρεί μια ολόκληρη επιφάνεια; Καλό ερώτημα...

Επιτρέψτε μας να σας εισαγάγουμε στον κόσμο των αντικειμένων και του αντικειμενοστρεφούς προγραμματισμού. Ναι, οι επιφάνειες αποτελούν αντικείμενα (objects). Τα αντικείμενα, με τη σειρά τους, αποτελούν ολόκληρες προγραμματιστικές δομές, οι οποίες διαθέτουν ιδιότητες και χαρακτηριστικά (attributes και properties), όπως επίσης και τις δικές τους συναρτήσεις. Οι συναρτήσεις ενός αντικειμένου ονομάζονται μέθοδοι και επιδρούν επάνω στις ιδιότητες και στα χαρακτηριστικά του ίδιου του αντικειμένου. Ο αντικειμενοστρεφής προγραμματισμός, λοιπόν, είναι εκείνος στον οποίο χρησιμοποιούνται αντικείμενα. Για την ώρα, είναι πολύ πιθανό να μην αντιλαμβάνεστε πλήρως την αξία και των αντικειμένων. Δεν πειράζει, θα την κατανοήσετε σιγά-σιγά. Ας δούμε μερικά παραδείγματα, ξεκινώντας με την παρακάτω γραμμή:

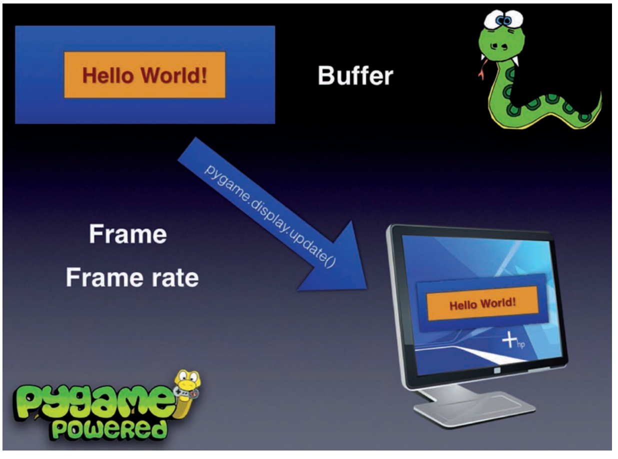
```
thetext = textfont.render("Hello World!", True, (255,0,0),(255,255,0))
```

Εδώ, μεταξύ άλλων, βλέπουμε το αντικείμενο textfont. Πρόκειται για ένα αντικείμενο τύπου font ή, όπως λέμε στον αντικειμενοστρεφή προγραμματισμό, για ένα αντικείμενο της κλάσης font. Δίπλα από το όνομα του αντικειμένου εμφανίζεται η συνάρτηση render. Αυτή η συνάρτηση δεν είναι γενικής χρήσης κι είναι αδύνατο να την καλέσουμε μόνη της. Συγκεκριμένα, χρησιμοποιείται μόνο σε συνδυασμό με τα αντικείμενα της κλάσης font. Με άλλα λόγια, αποτελεί μια μέθοδο των αντικειμένων της κλάσης font. Όπως βλέπετε και στον κώδικα, οι μέθοδοι καλούνται με τον εξής τρόπο:

```
αντικείμενο.ΌνομαΜεθόδου(παράμετροι_αν_υπάρχουν)
```

Στη γραμμή του κώδικα την οποία εξετάζουμε, αριστερά του ίσον εμφανίζεται το αντικείμενο thetext. Αναρωτιέστε πώς γίνεται να εξισώνουμε ένα αντικείμενο-επιφάνεια (κλάση surface), με ένα αντικείμενο-font; Είναι πολύ απλό: Η μέθοδος render δημιουργεί κι επιστρέφει ένα αντικείμενο της κλάσης surface! Αυτό το αντικείμενο περιέχει το μήνυμα, το οποίο έχουμε δώσει σαν παράμετρο στη μέθοδο render. Η όψη του μηνύματος καθορίζεται από τα χρώματα, τα οποία έχουμε δώσει (επίσης) ως παραμέτρους στη μέθοδο

Μια εικόνα αξίζει όσο χίλιες λέξεις. Δείτε με ένα απλό σχήμα πώς λειτουργεί το buffering. Η οθόνη σχεδιάζεται στην προσωρινή μνήμη και με το update μεταφέρεται στη μνήμη οθόνης.



render, αλλά και από τα γενικότερα χαρακτηριστικά του αντικειμένου thetext (γραμματοσειρά, μέγεθος, κ.λπ). Για περισσότερα περί μεθόδων κι αντικειμενοστρεφούς προγραμματισμού, δείτε το άρθρο 3.5 (http://deltahacker.gr/?p=5124).

Επιστροφή στο πρόγραμμα

Τώρα που αναφέραμε τα βασικά για τα αντικείμενα και τις μεθόδους, μπορούμε να συνεχίσουμε την ανάλυση του προγράμματος με άλλον αέρα. Ας δούμε πού είχαμε σταματήσει:

```
textx = centerMessage(thetext)
texty = 40
```

Αυτές οι δύο μεταβλητές θα χρησιμοποιηθούν αργότερα, για την προβολή της επιφάνειας thetext σε κάποια συγκεκριμένη θέση. Πρακτικά, οι συγκεκριμένες μεταβλητές θα αποτελέσουν τις συντεταγμένες, στις οποίες θα εμφανιστεί το αντικείμενο thetext. Όπως καταλαβαίνετε, σ' ένα παράθυρο γραφικών οι συντεταγμένες μετρώνται σε pixel. Αντίθετα, οι συντεταγμένες στην κονσόλα κειμένου μετρώνται σε στήλες και σε γραμμές χαρακτήρων. Στο πρόγραμμά μας, λοιπόν, το texty θα καθορίσει τη γραμμή (το ύψος) εμφάνισης του αντικειμένου, ενώ το textx θα καθορίσει τη στήλη (τη θέση στην οριζόντια διεύθυνση). Στο texty αποδίδουμε αυθαίρετα μια τιμή, η οποία θα εμφανίσει το αντικείμενο σε ικανοποιητικό ύψος. Για το textx, όμως, κάνουμε κάτι πιο πονηρό. Το αντικείμενο thetext θέλουμε να εμφανιστεί στη μέση του παραθύρου, ως προς την οριζόντια διεύθυνση. Έτσι, χρησιμοποιούμε τη συνάρτηση centerMessage. Αυτή η συνάρτηση λαμβάνει υπόψη της το μήκος του thetext, όπως επίσης το μήκος του παραθύρου του προγράμματος κι επιστρέφει τη θέση στην οποία πρέπει να τοποθετηθεί το thetext, ώστε να βρίσκεται στη μέση.

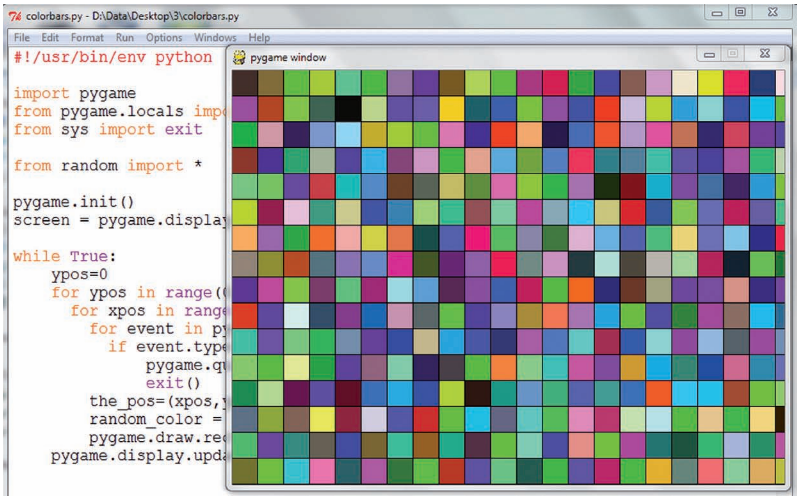
Όσοι δεν φάγατε το σοκολατάκι, μπορείτε να το φάτε τώρα. Όσοι το φάγατε, κρύψτε το κουτί για να μην τα φάτε όλα. Θα πάθετε ζάχαρο...

```
endprogram = False
while not endprogram:
      screen.fill(surfacecolor)
      screen.blit(thetext,(textx,texty))
      endprogram = getQuit()
      pygame.display.update()
pygame.quit()
exit()
```

Φτάσαμε στον κύριο βρόχο του προγράμματος! Ας δούμε πρώτα τις εντολές τις οποίες περιλαμβάνει και θα καταλάβετε μετά γιατί μιλάμε για βρόχο, σ' ένα πρόγραμμα που τυπώνει μόνο ένα μήνυμα.

```
screen.fill(surfacecolor)
```

Αν έχετε πιάσει το νόημα, θα φαντάζεστε ήδη ότι τα αντικείμενα τύπου επιφάνειας διαθέτουν τη μέθοδο fill, με την οποία μπορούμε να τα γεμίζουμε χρώμα. Έτσι, με τη συγκεκριμένη γραμμή κώδικα γεμίζουμε την επιφάνεια screen (ολόκληρο το παράθυρο του προγράμματός μας) με το χρώμα surfacecolor.



Η εικόνα που σχηματίζει το colorbars.py. Αφού το κατεβάσετε και το τρέξετε, σας προτείνουμε να αρχίσετε τις τροποποιήσεις. Δοκιμάστε να αλλάξετε το μέγεθος των τετραγώνων, τις αποστάσεις μεταξύ τους ή και το σχήμα τους. Προσπαθήστε να φτιάξετε παραλληλόγραμμα ή και κύκλους. Για βοήθεια μπορείτε να ανατρέξετε στο `www.pygame.org/docs/ref/draw.html`.

Στη συνέχεια, βλέπουμε μια ακόμα μέθοδο των αντικειμένων της κλάσης surface. Είναι η μέθοδος blit, η οποία επιτρέπει να τοποθετήσουμε ένα αντικείμενο πάνω σε ένα άλλο. Δείτε ξανά τη σχετική γραμμή:

```
screen.blit(thetext,(textx,texty))
```

Με τη βοήθεια της blit βάζουμε την επιφάνεια thetext πάνω στην επιφάνεια screen. Μάλιστα η επιφάνεια thetext τοποθετείται στις συντεταγμένες textx και texty. Παρατηρήστε ότι οι συντεταγμένες δίνονται σε μορφή tuple κι ακριβώς γι' αυτό το ζεύγος των textx και texty βρίσκεται μέσα σε παρενθέσεις.

```
pygame.display.update()
```

Αυτό κι αν είναι μυστήριο! Σε ποιον κάνουμε update και γιατί; Τα πράγματα είναι απλά: Οτιδήποτε κάνουμε σε οποιαδήποτε επιφάνεια, εμφανίζεται στην οθόνη μόνον όταν καλέσουμε την update. Αυτό συμβαίνει γιατί οι επιφάνειες τις οποίες δημιουργούμε, μαζί με όλα τα χαρακτηριστικά τους, βρίσκονται σε μια προσωρινή περιοχή της μνήμης (buffer). Για να δούμε στην οθόνη αυτές τις επιφάνειες πρέπει προηγουμένως τα περιεχόμενα της προσωρινής μνήμης να μεταφερθούν στη μνήμη της κάρτας γραφικών! Με άλλα λόγια, το πρόγραμμα σχηματίζει κάθε φορά ένα ολόκληρο καρέ και το τοποθετεί στη buffer. Για να δούμε αυτό το καρέ, πρέπει να το μεταφέρουμε από τη buffer στη μνήμη της κάρτας. Με αυτόν τον τρόπο, ο χρήστης της εφαρμογής δεν θα δει ποτέ τα περιεχόμενα της οθόνης να σχεδιάζονται σιγά-σιγά, αφού η αντιγραφή από τη μία μνήμη στην άλλη πραγματοποιείται ταχύτατα! Τώρα, πιστεύουμε ότι καταλαβαίνετε το ρόλο του DOUBLEBUF, το οποίο συναντήσαμε νωρίτερα στο πρόγραμμα. Η παρουσία του επιβάλει τη χρήση δύο buffer, για την αποθήκευση και την εμφάνιση όσων σχεδιάζουμε.

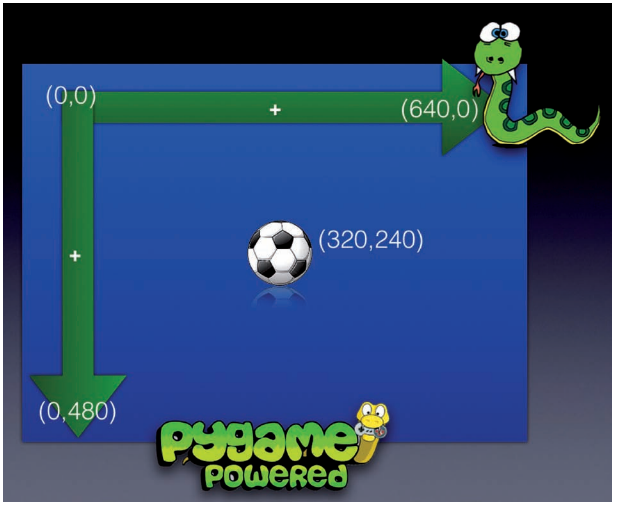
Event Driven Programming

Αν έχετε γράψει έστω κι ένα πρόγραμμα για το παραθυρικό περιβάλλον (ακόμα και σε Visual Basic, η οποία κρύβει πολλές λεπτομέρειες) θα έχετε συναντήσει τα λεγόμενα συμβάντα (events). Οι εφαρμογές για την κονσόλα δεν εμπεριέχουν την έννοια του συμβάντος: Η μια εντολή εκτελείται μετά την άλλη κι ο χρήστης μπορεί να παρέμβει στην εξέλιξη

του προγράμματος μόνον όταν υπάρχει κάποια εντολή του τύπου input -- ή έστω κάποια άλλη η οποία εκτελείται επαναλαμβανόμενα και ελέγχει διαρκώς το πληκτρολόγιο. Στις παραθυρικές εφαρμογές τα πράγματα είναι διαφορετικά. Ο χρήστης μπορεί να παρέμβει ανά πάσα στιγμή στη ροή ενός προγράμματος, ενώ το πρόγραμμα δεν ελέγχει διαρκώς τις "κινήσεις" του χρήστη. Αντίθετα, το πρόγραμμα βρίσκεται σε ετοιμότητα να δεχτεί και να επεξεργαστεί μηνύματα του συστήματος, τα οποία ούτε λίγο ούτε πολύ λένε κάτι τέτοιο: "Μόλις συμβεί το τάδε γεγονός, κάνε κάτι!". Για παράδειγμα, σκεφτείτε την περίπτωση κατά την οποία ο χρήστης κάνει κλικ στο κουμπάκι close (X) του παραθύρου. Ίσως πιστεύετε ότι αυτά τα κουμπιά τα χειρίζεται αποκλειστικά το λειτουργικό σύστημα. Τα πράγματα όμως δεν είναι έτσι. Το λειτουργικό σύστημα στέλνει μόνο ένα μήνυμα στην εφαρμογή, το οποίο λέει κάτι τέτοιο: "Ο χρήστης πάτησε το close". Προφανώς, η εφαρμογή πρέπει να είναι ανά πάσα στιγμή έτοιμη ώστε να επεξεργαστεί ένα τέτοιο μήνυμα και να αντιδράσει κατάλληλα. Σε αντίθετη περίπτωση, αν ο χρήστης συνεχίσει να πατάει με μανία το close τότε το λειτουργικό θα θεωρήσει ότι η εφαρμογή έχει κολλήσει, θα εμφανίσει το μήνυμα "Η εφαρμογή δεν αποκρίνεται" και θα την τερματίσει με τη βία! Όπως καταλαβαίνετε, τα προγράμματα για το παραθυρικό περιβάλλον οφείλουν να περιμένουν και να επεξεργάζονται διάφορα μηνύματα αυτού του είδους. Ουσιαστικά, αυτά τα μηνύματα αποτελούν ειδοποιήσεις για την πραγματοποίηση ενός event. Για παράδειγμα, μια παραθυρική εφαρμογή πρέπει να δέχεται τα μηνύματα που προέρχονται από το πληκτρολόγιο, ώστε ν' αντιδρά κατάλληλα στα κουμπιά που πατάει ο χρήστης. Αντίστοιχα, πρέπει να δέχεται τα μηνύματα που προέρχονται από το ποντίκι, ώστε ν' αποκρίνεται στα κλικ του χρήστη. Με άλλα λόγια, καθετί που κάνει ο χρήστης αποτελεί ένα event. Μια παραθυρική εφαρμογή πρέπει να 'ναι προετοιμασμένη για αυτά τα συμβάντα και να τα επεξεργάζεται κατάλληλα.

Το πρόγραμμά μας, λοιπόν, συνεχίζει να εκτελείται και μετά την εμφάνιση του μηνύματος "Hello World!". Αν ολοκληρωνόταν εκεί, το παράθυρο θα έκλεινε χωρίς να προλάβουμε να το δούμε! Μετά απ' όλα αυτά, μπορούμε να εξετάσουμε ένα κομμάτι του κώδικα το οποίο νωρίτερα είχαμε παραβλέψει:

Το σημείο (0,0) βρίσκεται στην πάνω αριστερή γωνία της οθόνης. Όσο κατεβαίνουμε και όσο κινούμαστε προς τα δεξιά, οι συντεταγμένες αυξάνονται. Σ' ένα παράθυρο μεγέθους 640X480, ένα αντικείμενο που εμφανίζεται στη μέση θα βρισκόταν στις συντεταγμένες (320,240).




```
def getQuit():
    for event in pygame.event.
get():
      if event.type == QUIT:
        return True
```

Όλα τα συμβάντα τα οποία λαμβάνει το πρόγραμμά μας καταλήγουν σε μια ουρά (queue). Η pygame.event.get() δημιουργεί μια λίστα με τα συμβάντα αυτής της ουράς και με τη βοήθεια του βρόχου for, τα ελέγχουμε ένα προς ένα. Έτσι, η συνάρτηση defQuit ελέγχει όσα events έχουν προκύψει κι αν εντοπίσει κάποιο του τύπου QUIT, επιστρέφει True. Η συνάρτηση defQuit χρησιμοποιείται μέσα στον κύριο βρόχο του προγράμματός μας, ως εξής:

```
endprogram = getQuit()
```

Όπως καταλαβαίνετε, η μεταβλητή endprogram γίνεται True όταν τη συνάρτηση getQuit επιστρέψει True. Δηλαδή, η συγκεκριμένη μεταβλητή γίνεται True όταν εντοπιστεί ένα event του τύπου Quit. Όμως η τιμή αυτής της μεταβλητής χρησιμοποιείται σαν κριτήριο για την επανάληψη του κύριου βρόχου του προγράμματος. Έτσι, όταν εντοπιστεί ένα συμβάν του τύπου Quit, το τελικό αποτέλεσμα θα είναι ο τερματισμός αυτού του βρόχου. Έξω από το βρόχο, το πρόγραμμά μας τερματίζει με τις ακόλουθες γραμμές:

```
pygame.quit()
exit()
```

Τέλος, σημειώστε ότι η επεξεργασία των event πρέπει να γίνεται πολύ συχνά, ώστε το πρόγραμμα να αποκρίνεται ταχύτατα σε όσα συμβαίνουν. Σε διαφορετική περίπτωση, θα μοιάζει να σέρνεται.

Frame και Framerate

Οι πιο παρατηρητικοί από εσάς θα είδατε ότι το μήνυμα “Hello World!” τυπώνεται συνέχεια, μέσα στο βρόχο while. Είναι απαραίτητο αυτό; Πόσες φορές το δευτερόλεπτο συμβαίνει; Τι θα γίνει αν μετακινήσουμε τις εντολές εμφάνισης έξω από το βρόχο και αφήσουμε μόνο την επεξεργασία των συμβάντων στο εσωτερικό του;

Στο πρόγραμμά μας, κάθε φορά που εκτελείται ο βρόχος εμφανίζεται το ίδιο, στατικό μήνυμα. Σε ένα παιχνίδι, όμως, κάθε εκτέλεση του βρόχου θα ήταν πολύ πιθανό να περιέχει αρκετές αλλαγές. Για παράδειγμα, θα μπορούσε να περιέχει τη μετατόπιση διαφόρων αντικειμένων, την εμφάνιση νέων, την εξαφάνιση κάποιων άλλων κ.ο.κ. Σε μια τέτοια περίπτωση, η εικόνα που θα εμφάνιζε κάθε επανάληψη του βρόχου θ’ αποτελούσε ένα στιγμιότυπο των κινήσεων. Με απλά λόγια, θα ήταν ένα καρέ (frame). Όπως ξέρετε --ειδικά όσοι παίζετε με first person shooters-- όσο πιο πολλά καρέ παράγονται σ’ ένα δευτερόλεπτο, τόσο πιο ομαλή είναι η κίνηση του παιχνιδιού. Τυπικά, για να έχουμε *τέλεια* κίνηση χρειαζόμαστε 50 καρέ το δευτερόλεπτο. Ο ρυθμός των καρέ ονομάζεται framerate και είναι επίσης γνωστός, σε όσους παίζουν παιχνίδια FPS.

Το δικό μας πρόγραμμα, αν και εμφανίζει κάτι στατικό, δεν παύει να έχει ένα κάποιο framerate. Αναρωτιέστε πόσο είναι; Το πρόγραμμά μας εμφανίζει τόσα καρέ το δευτερόλεπτο, όσα αντέχει ο υπολογιστής μας. Στο pygame μπορούμε να μετρήσουμε το framerate που πετυχαίνει το πρόγραμμά μας, όπως επίσης και να το περιορίσουμε. Όλα αυτά, αν και ακούγονται περιττά, θα δείτε ότι είναι πολύ χρήσιμα.

Τέλος, μένει ν’ απαντήσουμε στο ερώτημα για την τοποθέτηση των εντολών εκτός του βρόχου. Όπως καταλαβαίνετε, κάτι τέτοιο θα ήταν απόλυτα εφικτό και δεν θα επηρέαζε καθόλου τη συγκεκριμένη εφαρμογή μας. Σε ένα παιχνίδι, όμως, θα ήταν μοιραίο: Τίποτα δεν θα κουνιόταν και η εφαρμογή ουσιαστικά θα ήταν κολλημένη.

Γραφικά και κίνηση

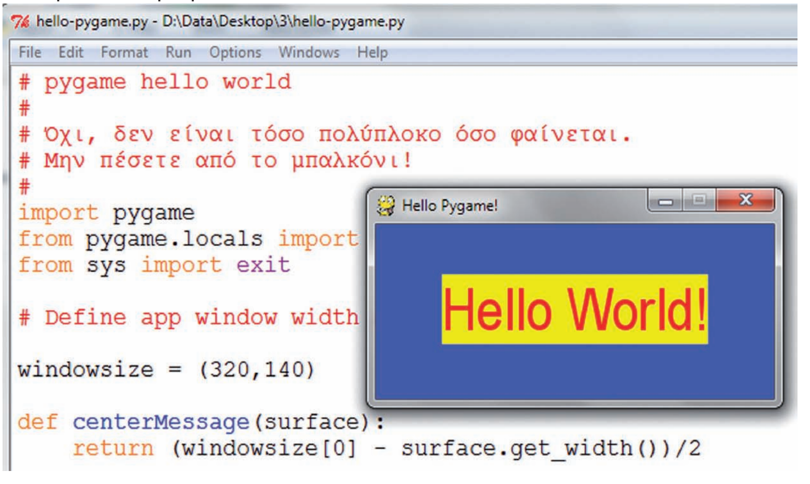
Ας δούμε επιτέλους και λίγη δράση! Το πρόγραμμα που θα παρουσιάσουμε τώρα υλοποιεί το γνωστό παιχνιδάκι bouncing ball. Μπορείτε να κατεβάστε τον κώδικα από εδώ:

http://bit.ly/bouncingb

Στο πακέτο θα βρείτε και μια εικόνα, η οποία είναι απαραίτητη για τη λειτουργία του προγράμματος. Ας δούμε μερικά ενδιαφέροντα σημεία του κώδικα:

```
x,y = 100.0,100.0
xspeed,yspeed = 50,50
```

Με αυτές τις μεταβλητές ορίζουμε την αρχική θέση της μπάλας καθώς και την αρχική της ταχύτητα, σε κάθε άξονα. Όπως αντιλαμβάνεστε, έχοντας την ίδια ταχύτητα και στους δύο άξονες, η μπάλα θα κινηθεί διαγώνια και υπό γωνία 45 μοιρών.



Το hello-world σε pygame. Ναι, το ξέρουμε ότι δείχνει μεγάλο, αλλά κάνει πολύ περισσότερα πράγματα απ' όσα θα περίμενε κανείς. Αφήστε τις εικόνες κι αρχίστε να διαβάζετε το άρθρο, για να μάθετε τα πάντα!

```
ballimage = 'soccer-ball.png'
ball = pygame.image.load(ballimage)
```

Εδώ πραγματοποιείται η φόρτωση της εικόνας της μπάλας, από το αρχείο soccer-ball.png. Έτσι, δημιουργείται μια επιφάνεια (surface) η οποία ονομάζεται ball και περιέχει την εικόνα της μπάλας.

```
ballwidth = ball.get_width()
ballheight = ball.get_height()
```

Όπως βλέπετε, μπορούμε να ζητήσουμε από ένα αντικείμενο τύπου surface να μας πει τις διαστάσεις του! Αυτό γίνεται με τις μεθόδους get_width() και get_height().

```
clock = pygame.time.Clock()
```

Αυτή η γραμμή δημιουργεί ένα αντικείμενο τύπου clock, με το (πρωτότυπο) όνομα clock. Ας δούμε πώς χρησιμοποιείται. Μέσα στον κύριο βρόχο του προγράμματος, εμφανίζεται η ακόλουθη γραμμή:

```
time = clock.tick()
```

Στη μεταβλητή time αποθηκεύεται ο χρόνος που πέρασε από την προηγούμενη κλήση της μεθόδου tick(), σε χιλιοστά του δευτερολέπτου. Αυτή η γραμμή εκτελείται σε κάθε επανάληψη του βρόχου κι επομένως, μετράει το χρόνο που απαιτείται για την ολοκλήρωση μιας επανάληψης. Όμως σε κάθε επανάληψη του βρόχου εμφανίζεται ένα νέο frame. Έτσι, έχουμε ένα frame ανά χρόνο time. Αυτό μάς κάνει framerate ίσο με 1000/time. Σας φαίνεται παράξενη αυτή η σχέση; Περιμένατε να σας πούμε ότι το

framerate είναι ίσο με 1/time; Λογικός προβληματισμός, αλλά ξεχνάτε ότι το time είναι ένας χρόνος μετρημένος σε χιλιοστά του δευτερολέπτου. Αντίθετα, το framerate το μετράμε σε καρέ ανά δευτερόλεπτο. Κανονικά, λοιπόν, θα έπρεπε να μετατρέψουμε το time σε δευτερόλεπτα και να πούμε ότι το framerate είναι ίσο με 1/(time/1000). Αυτή η σχέση όμως, είναι ισοδύναμη με την framerate=1000/time. Σοκολατάκι, κανείς; :)

Στο πρόγραμμά μας η μπάλα θέλουμε να κινείται με την ίδια ταχύτητα, ανεξάρτητα από τις επιδόσεις του υπολογιστή. Για να το πετύχουμε αυτό, πρέπει η μπάλα να διανύει σταθερή απόσταση στη μονάδα του χρόνου (π.χ, στο 1sec). Έτσι, σ' ένα γρήγορο μηχάνημα η κίνηση θα είναι ομαλή, ενώ σ' ένα αργό θα είναι σπασμωδική και η μπάλα θα κάνει πηδηματάκια! Πάντως, σε κάθε περίπτωση η μπάλα θα κινείται με την ίδια ταχύτητα. Μήπως σας μπέρδεψε η παραπάνω εξήγηση; Θυμηθείτε λίγο τη φυσική[2]. Η εξίσωση για την ευθύγραμμη ομαλή κίνηση είναι:

Ταχύτητα = Απόσταση / Χρόνος

Επομένως:

Απόσταση = Ταχύτητα * Χρόνος

Αυτή η εξίσωση θα μας χρησιμεύσει και στο πρόγραμμα. Πριν ξεκινήσουμε τις πράξεις, όμως, θα πρέπει να κάνουμε μια μετατροπή. Η time διατηρεί το χρόνο εκτέλεσης του βρόχου σε χιλιοστά του δευτερολέπτου. Ωστόσο εμείς συνηθίζουμε να δουλεύουμε με δευτερόλεπτα. Επομένως, για να πάρουμε το χρόνο εκτέλεσης σε δευτερόλεπτα, θα διαιρέσουμε την τιμή της time με το χίλια:

time = time / 1000.0

Ωραία. Τώρα μπορούμε να υπολογίσουμε την απόσταση που πρέπει να διανύει η μπάλα, σε κάθε επανάληψη του βρόχου. Αυτό θα γίνει με τη βοήθεια του παραπάνω τύπου:

distance_x = time * xspeed
distance_y = time * yspeed

Γνωρίζοντας την απόσταση που πρέπει να διανύσει η μπάλα σε κάθε άξονα, μπορούμε εύκολα να υπολογίσουμε και τη νέα της θέση:

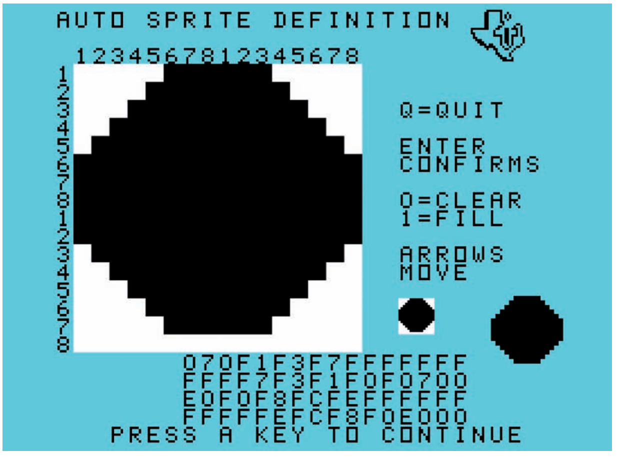
x = x + distance_x
y = y + distance_y

Το πρόγραμμα συνεχίζει με τις ακόλουθες γραμμές:

```
if (x > (640.0-ballwidth) or x<=0.0):
    xspeed = -xspeed
if (y > (480.0-ballheight) or y<=0.0):
    yspeed = -yspeed
```

Όπως καταλαβαίνετε, με αυτά τα if ελέγχουμε αν η μπάλα έχει φτάσει τα όρια του παραθύρου. Εφόσον

Την παλιά, καλή εποχή, δεν υπήρχε το Internet για να κατεβάσει κανείς τη μπάλα της αρεσκείας του, σε έτοιμο bitmap. Τα γραφικά σχεδιάζονταν στο χέρι (σε χαρτί με τετραγωνάκια!) ή σε προγράμματα που στις μέρες μας θα έμοιαζαν με icon editors. Φυσικά, εννοείται ότι και αυτά τα προγράμματα τα γράφαμε μόνοι μας! Στην εικόνα βλέπετε το Auto Sprite Definition. Πρόκειται για τη δική μου εκδοχή ενός προγράμματος της Texas Instruments, το οποίο επιτρέπει τη σχεδίαση γραφικών (sprites). Άπειρα pacman και διαστημόπλοια έχουν σχεδιαστεί σε αυτό το πρόγραμμα!



2. Τελικά η φυσική δεν είναι μόνο για να λύνετε ασκήσεις. Είναι και για να γράφετε παιχνίδια, αν κι αυτό θα παρέλειψαν να σας το πουν οι καθηγητές σας...

συμβεί κάτι τέτοιο, η ταχύτητά της ως προς τον σχετικό άξονα αντιστρέφεται. Έτσι, αντιστρέφεται και η φορά κίνησης!

Το παιχνιδάκι μας τυπώνει στην οθόνη και το framerate. Αυτό γίνεται με την επιφάνεια thetext, η οποία ενσωματώνεται στο παράθυρο της εφαρμογής με τη βοήθεια της μεθόδου blit:

```
thetext = textfont.render(str(1000/time), True, (255,0,0),(255,255,0))
screen.blit(thetext,(0,0))
```

Όσο πιο γρήγορη είναι η CPU και η κάρτα γραφικών του υπολογιστή σας, τόσο μεγαλύτερο θα είναι και το framerate που θα δείτε. Πάντως, μπορείτε αν θέλετε να τον περιορίσετε. Για τον σκοπό αυτό, αφαιρέστε το χαρακτήρα των σχολίων από την παρακάτω γραμμή:

```
framerate = 30
```

Μετά αλλάξετε τον κώδικα μέσα στον βρόχο, ώστε η γραμμή με τη μέθοδο tick να γίνει έτσι:

```
time = clock.tick(framerate)
```

Με αυτή τη σύνταξη η γραμμή time θα προκαλεί όση καθυστέρηση χρειάζεται, ώστε το πρόγραμμά μας να επιτυγχάνει το συγκεκριμένο framerate! Αν θέλετε, δοκιμάστε και με διαφορετικούς αριθμούς για να δείτε το αποτέλεσμα. Πάντως με περισσότερα από 60 καρέ το δευτερόλεπτο ενδέχεται να μην βλέπετε καμία αισθητή διαφορά. Σε χαμηλά framerate, όμως, η διαφορά θα είναι εμφανής και η ποιότητα κίνησης της μπάλας δεν θα είναι καλή.

Σαν άσκηση, δοκιμάστε το πρόγραμμα με διαφορετικές ταχύτητες μπάλας και framerates. Μετατρέψτε το ώστε να κινούνται δύο μπάλες αντί για μία! (Hint: Μπορείτε να κάνετε blit το ίδιο surface σε δύο διαφορετικές θέσεις.) Όταν οι δύο μπάλες συναντιούνται στην οθόνη, κάποια περνάει πάνω από την άλλη. Ποια και γιατί; Μπορείτε να το βρείτε;

Σας αφήνουμε τώρα να παίξετε και σας προκαλούμε να γράψετε τα δικά σας προγραμματάκια με animation. Επιπρόσθετα, μπορείτε να κατεβάσετε και να μελετήσετε ένα ακόμα δικό μας πρόγραμμα, από εδώ:

http://bit.ly/thecolorbars

Στο επόμενο τεύχος θα γράψουμε το πρώτο μας κανονικό παιχνιδάκι και θα μάθουμε ταυτόχρονα πώς γίνεται η ανάγνωση του πληκτρολογίου. Φυσικά, μην παραλείψετε να διαβάσετε και το συνοδευτικό άρθρο 3.5, στο http://deltahacker.gr.

Καλά ξενύχτια!



Πλήρη άρθρα που συμπληρώνουν το περιοδικό. Τα εβδομαδιαία μας video casts [έρχονται, έρχονται]. Ολόκληρα blasts from the past. Τα βίντεο από τα live streams. Τα πάντα ενδιαφέροντα forums. Το σημείο συνάντησης για τους φίλους του περιοδικού. Η πιο ζωντανή κοινότητα και οι πιο ανήσυχοι, δημιουργικοί χρήστες!

deltahacker.gr

What have you hacked for me lately?

ΣΑΡΚΟΒΟΡΟ για τα malware



by Ion

Skill: Advanced

Tags: Dionaea, conficker, phpLiteAdmin, GnuPlotSQL

Στο άρθρο που αρχίζει από τη σελίδα 40 του τεύχους παρουσιάσαμε το Dionaea, ένα εξαιρετικά αποτελεσματικό malware honeypot για Linux. Η εγκατάσταση και ρύθμισή του δεν είναι ακριβώς ό,τι θα χαρακτηρίζαμε "παιχνιδάκι" (Σ.τ.Ε. ούτε κομμάτι από κέικ). Το θέμα είναι ότι για το Dionaea και τις εξαρτήσεις του δεν υπάρχουν έτοιμα πακέτα, ενώ σ' ορισμένες περιπτώσεις το honeypot απαιτεί πολύ συγκεκριμένες εκδόσεις βιβλιοθηκών για να λειτουργήσει σωστά. Δείξαμε, μολαταύτα, αναλυτικά και βήμα προς βήμα την εγκατάσταση, τη ρύθμιση και την ενεργοποίηση του εκπληκτικού αυτού malware. Νομίζουμε ότι είναι ώρα να το δούμε και σε δράση.

Στο δικό μας σύστημα, οι πρώτες επιθέσεις από malware άρχισαν να εμφανίζονται μέσα σε λίγα λεπτά. Ας δούμε κάποιους τρόπους παρακολούθησης της κίνησης στο honeypot. Ένα πρώτο τρικ είναι η παρακολούθηση των αρχείων καταγραφής του Dionaea -- ή ακόμα και του ίδιου του p0f. Για το σκοπό αυτό, μπορούμε να χρησιμοποιήσουμε το προγραμματάκι tail με την παράμετρο –f:

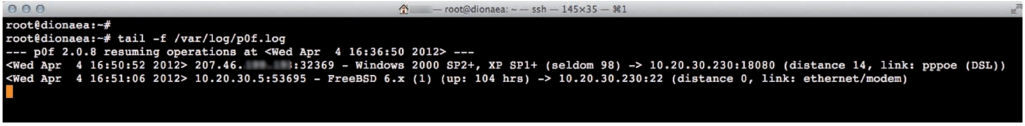
```
root@dionaea:~# tail -f /var/log/p0f.log
```

Με την παραπάνω γραμμή παρακολουθούμε σε πραγματικό χρόνο όσα καταγράφει το p0f. Τελικά, δεν αργήσαμε πολύ να κατεβάσουμε το πρώτο μας malware! Όπως είπαμε, όλα τα αρχεία αποθηκεύονται στον κατάλογο /opt/dionaea/var/dionaea/binaries. Τα περισσότερα από τα αρχεία που συλλέξαμε ήταν εκτελέσιμα σε περιβάλλον Windows. Το Dionaea αποστέλλει κάθε τέτοιο αρχείο σε online υπηρεσίες για malware analysis, όπως αναφέραμε και νωρίτερα. Έτσι, σε σύντομο χρονικό διάστημα, πήραμε ένα σχετικό email από το Norman Sandbox:

```
[ DetectionInfo ]
    * Filename: C:\analyzer\scan\64a8a34f7fcec9943132c5
8a89321257.
    * Sandbox name: NO_MALWARE
    * Signature name: W32/Conficker.EJ.
    * Compressed: NO.
    * TLS hooks: NO.
    * Executable type: Library(DLL).
    * Executable file structure: OK.
    * Filetype: PE_I386.
 [ General information ]
    * File length: 168480 bytes.
    * MD5 hash: 64a8a34f7fcec9943132c58a89321257.
    * SHA1 hash: f90e039a28dc045a5e41b86379e894d23f4df7a6.
    * Entry-point detection: Microsoft Visual C++ 6.0 DLL.
```

Το πρώτο μας malware, λοιπόν, αφορά σ' ένα variant του γνωστού worm Conficker. Το συγκεκριμένο worm θα σας απασχολήσει αρκετά. Αν αφήσετε το honeypot online για μερικές μέρες, θα παρατηρήσετε ότι το Conficker θα κάνει μεγάλο χαμό, παρά τις επίπονες προσπάθειες μεγάλων εταιρειών να σταματήσουν την εξάπλωση του. Μιλάμε για ένα worm το οποίο έχει υπό τον έλεγχό του χιλιάδες (αν όχι εκατομμύρια) μηχανήματα! Αν θέλετε να διαβάσετε περισσότερα γι' αυτό, μπορείτε να επισκεφθείτε την ιστοσελίδα του Conficker Working Group: www.confickerworkinggroup.org. Πρόκειται για μια ειδική ομάδα, η οποία δημιουργήθηκε με μοναδική αποστολή την καταπολέμηση του Conficker και στην οποία συμμετέχουν οργανισμοί όπως ο ICANN και εταιρείες όπως Microsoft, Cisco, McAfee, ESET, F-Secure κ.ά.

Είναι απίστευτο. Το Dionaea έχει δεν έχει 5 λεπτά που λειτουργεί σ' ένα από τα VM των δοκιμών μας, και ήδη το επισκέφτηκε ένα malware από κάποιο μολυσμένο Windows box! Η δεύτερη σύνδεση που βλέπετε, από κάτω, είναι από τον SSH client στον iMac του έκπληκτου χειριστή...





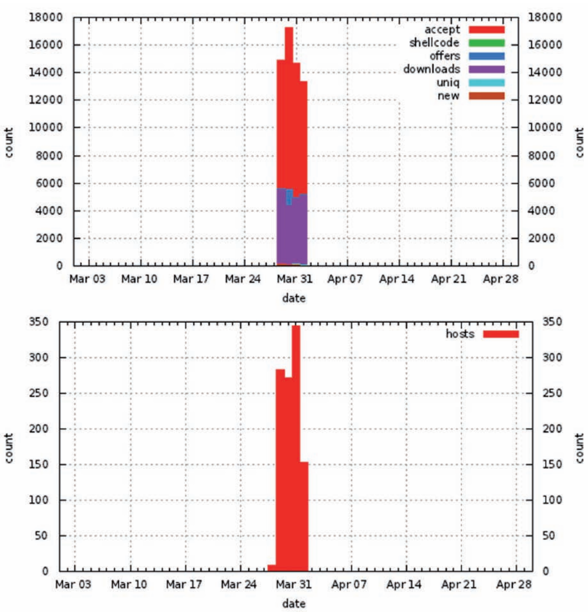
Στην εικόνα φαίνεται η κεφαλή της σελίδας με τα αποτελέσματα, από την υπηρεσία Anubis. Κάτι τέτοιο έφτασε στο email μας, όταν το Dionaea έστειλε στην εν λόγω υπηρεσία το συλληφθέν worm!

Μετά από τη σύλληψη του πρώτου μας malware, αφήσαμε το honeypot ενεργό για τις επόμενες τρεις μέρες. Πρόκειται για ένα ασφαλές χρονικό διάστημα, από το οποίο μπορούμε να πάρουμε μια καλή γεύση από τη δραστηριότητα των malware. Ας δούμε κάποια στοιχεία από τη λειτουργία της παγίδας μας, καθώς και μερικά πολύ χρήσιμα script, τα οποία θα μας βοηθήσουν να αναλύσουμε τ' αποτελέσματα.

Μέσα στις 80+ ώρες λειτουργίας του honeypot, η βάση δεδομένων του Dionaea έφτασε τα 58MB σε μέγεθος και το σύστημα κατέβασε περίπου 57MB αρχείων malware. Σας μιλάμε για σχεδόν 260 ξεχωριστά δείγματα, τα οποία αποθηκεύτηκαν αλλά και στάλθηκαν για ανάλυση. Για να πετύχουμε μια πρώτη στατιστική ανάλυση των καταγεγραμμένων δεδομένων, προσφέρεται ένα script σε Python με το όνομα mimic-nepstats.py. Μπορούμε να το κατεβάσουμε από τη διεύθυνση `http://bit.ly/Hmbfeq` και να το τρέξουμε στο σύστημά μας, ως εξής:

```
root@dionaea:~# wget http://bit.ly/Hmbfeq -O mimic-nepstats.py
root@dionaea:~# python mimic-nepstats.py
Statistics engine written by Andrew Waite - www.infosanity.co.uk
Number of submissions: 19874
Number of unique samples: 260
Number of unique source IPs: 602
First sample seen: 2012-03-28 23:10:25.210700
Last sample seen: 2012-04-01 12:15:40.302722
System Uptime: 3 days, 13:05:15.092022
Average daily submissions: 6624
Most recent submissions:
[...]
```

Όπως βλέπουμε, το σκριπτάκι παρέχει μερικά απλά στατιστικά στοιχεία γύρω από τη λειτουργία του honeypot. Το number of submissions αντιστοιχεί στο πλήθος των λήψεων, τις οποίες πραγματοποίησε το σύστημά



Κάθε φορά που ζητάμε από το gnuplotsql να σχεδιάσει ένα διάγραμμα, παράγεται κι ένα ακόμα. Πρόκειται για το default διάγραμμα που απεικονίζει τη συνολική δραστηριότητα του honeypot. Στην εικόνα φαίνεται το βλέπουμε μετά από τρεις μέρες διαρκούς λειτουργίας του honeypot.

μας. Το number of unique samples δείχνει το πλήθος των μοναδικών/ξεχωριστών αρχείων malware, τα οποία κατέβασε το honeypot. Στη συνέχεια εμφανίζεται το πλήθος των μοναδικών διευθύνσεων IP από τις οποίες δεχτήκαμε επίθεση κ.ο.κ.

Εκτός απ' αυτό το script, ο ίδιος προγραμματιστής προσφέρει κι ένα ακόμα. Αυτό παρέχει αναλυτικότερα στοιχεία, πραγματοποιώντας μια σειρά ερωτημάτων προς τη βάση SQLite του Dionaea. Το εν λόγω script είναι επίσης γραμμένο σε Python και ονομάζεται dionaea-sqlquery-0_2.py. Μπορούμε να το κατεβάσουμε από τη διεύθυνση http://bit.ly/Hnnt5r:

```
root@dionaea:~# wget http://bit.ly/Hnnt5r -O dionaea-sqlquery-0_2.py
```

Με το dionaea-sqlquery-0_2 μπορούμε να εξάγουμε από τη βάση του Dionaea τα ακόλουθα έξι στοιχεία: Τη συχνότητα των επιθέσεων ανά θύρα, τις επιθέσεις κατά τη διάρκεια μιας ημέρας (ανά ώρα), τα "δημοφιλέστερα" ληφθέντα malware, τους επιτιθέμενους με τον μεγαλύτερο αριθμό συνδέσεων, καθώς και τις συνδέσεις των τελευταίων 24 ωρών. Για να μάθουμε κάτι από όλα αυτά αρκεί να τρέξουμε το dionaea-sqlquery-0_2 με παράμετρο το --query X, όπου στη θέση του X θα βάλουμε τον αριθμό του στοιχείου που μας ενδιαφέρει. Για παράδειγμα, για να δούμε τη συχνότητα των επιθέσεων ανά θύρα στο δικό μας σύστημα, δώσαμε αυτό:

```
root@dionaea:~# python dionaea-sqlquery-0_2.py --query 1
Description:
        List of attacked ports
Executed Query:
        SELECT COUNT(local_port) AS hitcount,local_port AS port FROM connections WHERE connection_type = "accept" GROUP BY local_port HAVING COUNT(local_port) > 10
('HitCount', 'port')
(45282, 445)
(259, 1433)
```

Όπως βλέπουμε, το honeypot έχει δεχτεί παραπάνω από 45.000 επιθέσεις/συνδέσεις στη θύρα 445 (SMB), καθώς κι άλλες 260 στη θύρα 1433 (Microsoft SQL Server).

Εκτός από τα παραπάνω χρήσιμα scripts, το Dionaea συνοδεύεται κι αυτό από μερικά --επίσης χρήσιμα-- Python scripts. Βρίσκονται στον κατάλογο εγκατάστασης του honeypot, στη διαδρομή /opt/dionaea/bin. Το πρώτο απ' αυτά ονομάζεται readlogsqltree. Όταν το εκτελούμε συνδέεται με τη

βάση δεδομένων του Dionaea κι εμφανίζει όλες τις πληροφορίες για κάθε επίθεση που δεχτήκαμε. Η έξοδος του συγκεκριμένου script ενδέχεται να σας φανεί ακατάληπτη αρχικά, αλλά όσο αποκτάτε μεγαλύτερη εμπειρία με το Dionaea θα διαπιστώνετε ότι το readlogsqltree είναι πολύ χρήσιμο. Αν θέλετε να μάθετε περισσότερα για τη λειτουργία και τις παραμέτρους που δέχεται, μπορείτε να ρίξετε μια ματιά και στον κώδικά του -- σας το προτείνουμε ανεπιφύλακτα! Χρησιμοποιώντας ορισμένες από τις παραμέτρους που δέχεται μπορούμε να ζητήσουμε πληροφορίες από τη βάση του Dionaea για οποιαδήποτε επίθεση -- ή και όποια άλλα στοιχεία θέλουμε. Για παράδειγμα, για να δούμε όλες τις επιθέσεις των τελευταίων έξι ωρών, θα αρκούσε να δώσουμε τα ακόλουθα:

```
root@dionaea:~# cd /opt/dionaea/bin

root@dionaea:/opt/dionaea/bin# ./python3.2 readlogsqltree -t
$(date '+%s')-6*3600 /opt/dionaea/var/dionaea/logsql.sqlite
```

Με το παρακάτω θα βλέπαμε τις επιθέσεις ενάντια στην υπηρεσία MSSQL, οι οποίες πραγματοποιήθηκαν τις τελευταίες δύο ώρες:

```
root@dionaea:/opt/dionaea/bin# ./python3.2 readlogsqltree -t
$(date '+%s')-2*3600 -P mssqld /opt/dionaea/var/dionaea/logsql.
sqlite
```

Οπτικοποίηση και πρόσβαση

Ένα ακόμα ενδιαφέρον script που παρέχεται μαζί με το Dionaea είναι το gnuplotsql. Κάτι μάς λέει ότι αυτό θα σας αρέσει περισσότερο! Σκοπός του είναι η δημιουργία διαγραμμάτων, για την παρουσίαση διαφόρων πληροφοριών. Όπως και το προηγούμενο script, το gnuplotsql βρίσκεται στον κατάλογο /opt/dionaea/bin. Ωστόσο για να το χρησιμοποιήσουμε θα πρέπει προηγουμένως να εγκαταστήσουμε την ακόλουθη βιβλιοθήκη λογισμικού:

```
root@dionaea:/opt/dionaea/bin# apt-get install –y gnuplot
```

Κατόπιν, μπορούμε να δημιουργήσουμε διαγράμματα -- π.χ., ανά πρωτόκολλο. Μαζί με κάθε διάγραμμα που ζητάμε, το script παράγει και ένα βασικό διάγραμμα με τη συνολική κατάσταση λειτουργίας και κίνησης δεδομένων στο honeypot. Για να υλοποιήσουμε το παράδειγμα που αναφέραμε, αρκεί να εκτελέσουμε την ακόλουθη γραμμή (αγνοώντας τα όποια warnings):

```
root@dionaea:/opt/dionaea/bin# ./python3.2 gnuplotsql -d \ /opt/
dionaea/var/dionaea/logsql.sqlite -p smbd -p mssqld -p ftpd
```

Η παραπάνω εντολή θα δημιουργήσει ένα διάγραμμα για καθένα από τα τρία πρωτόκολλα: SMB, MSSQL και FTP. Σημειώστε ότι τα διαγράμματα αποθηκεύονται στον κατάλογο /tmp/dionaea-gnuplot. Για να μπορέσουμε να τα δούμε, μπορούμε να τα μετακινήσουμε στο DocumentRoot του web server που έχουμε στο VPS μας:

```
root@dionaea:/opt/dionaea/bin# cd /tmp

root@dionaea:/tmp# cp -rf dionaea-gnuplot /var/www/
```

Στη συνέχεια, για να δούμε τα διαγράμματα αρκεί να επισκεφτούμε τη διεύθυνση `http://IP-του-honeypot/dionaea-gnuplot`. Αν δεν έχουμε κάποιον web server στο μηχάνημα με το honeypot, μπορούμε να καταφύγουμε σε μια λιγότερο εντυπωσιακή λύση: Κατέβασμα των διαγραμμάτων με secure copy (μέσω SSH, δηλαδή).

Τέλος, μπορούμε αν θέλουμε ν' αποκτήσουμε άμεση πρόσβαση στη βάση δεδομένων του Dionaea. Κάτι τέτοιο θα μπορούσε να γίνει από την κονσόλα, αλλά δεν θα ήταν καθόλου πρακτικό. Για καλή μας τύχη, υπάρχει ένα δωρεάν εργαλείο το οποίο θυμίζει το phpMyAdmin αλλά είναι φτιαγμένο για βάσεις τύπου SQLite. Πρόκειται για το phpLiteAdmin, το οποίο μπορούμε να εγκαταστήσουμε αρκετά εύκολα. Για τη λειτουργία του phpLiteAdmin απαιτούνται ωστόσο ορισμένα πρόσθετα πακέτα. Συγκεκριμένα, θα πρέπει να εγκαταστήσουμε τα ακόλουθα:

```
root@dionaea:~# apt-get install -y php5 php5-sqlite
root@dionaea:~# service apache2 restart
```

Κατόπιν μπορούμε να κατεβάσουμε το phpLiteAdmin και να το αποσυμπιέσουμε στο DocumentRoot του web server:

```
root@dionaea:~# cd /var/www
root@dionaea:/var/www# wget http://phpliteadmin.googlecode.com/
files/phpliteadmin_v1-9-1.zip
root@dionaea:/var/www# apt-get install unzip
root@dionaea:/var/www# unzip phpliteadmin_v1-9-1.zip
```

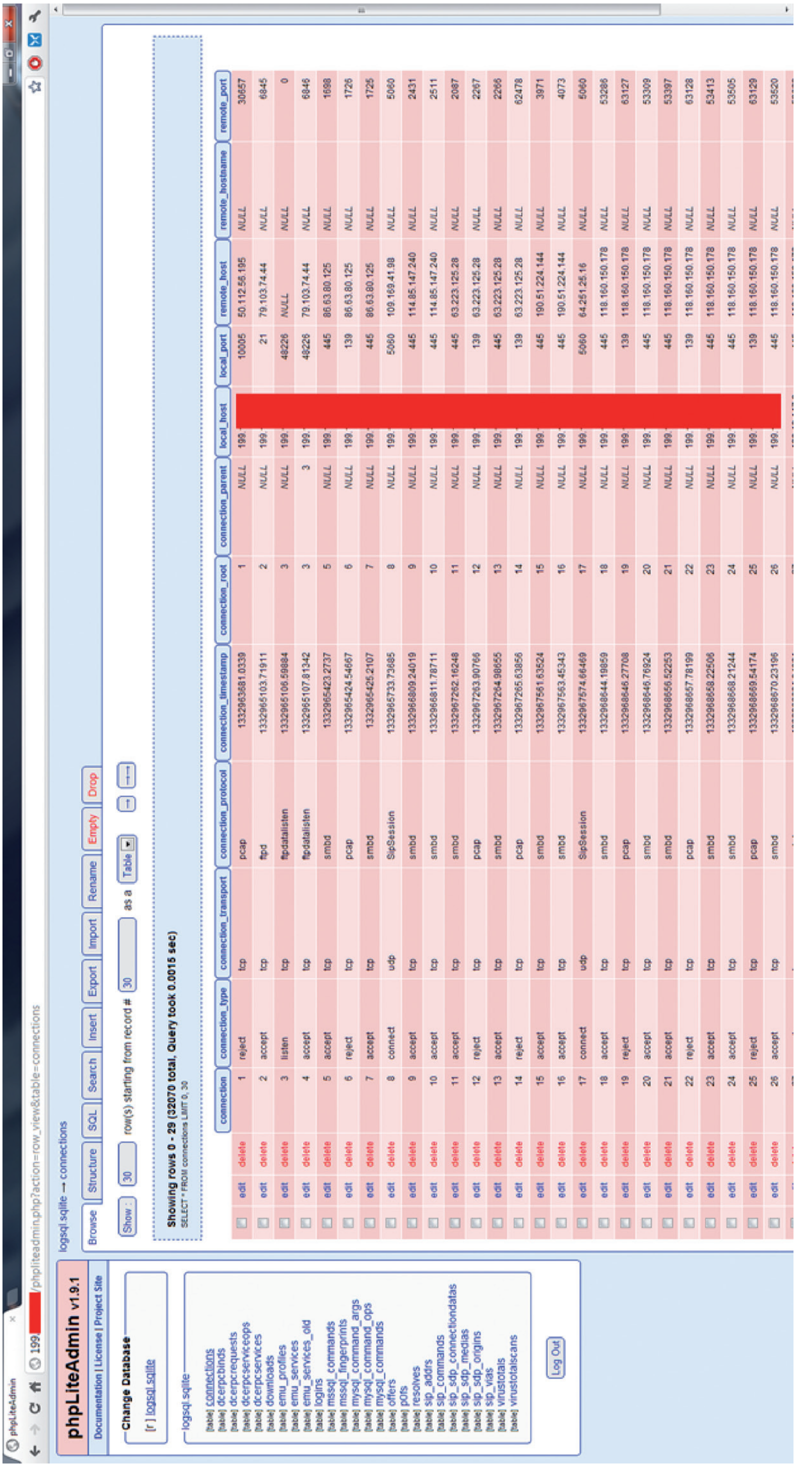
(Προφανώς, αν δεν έχετε εγκαταστήσει κάποιον web server καταλαβαίνετε ότι είναι ώρα να το κάνετε. Προτείνουμε έναν εκ των nginx, apache2. Πριν ξεκινήσετε τον web server τερματίστε όλα τα instances του Dionaea δίνοντας "killall dionaea", μετά ξεκινήστε τον web server, π.χ., με "service apache2 start" και, τέλος, ξεκινήστε και πάλι το Dionaea.) Πριν συνδεθούμε στο phpLiteAdmin από κάποιον web browser, πρέπει να κάνουμε μερικές απλές επεμβάσεις στο αρχείο phpliteadmin.php. Από εκεί θα πρέπει να καθορίσουμε τον κωδικό σύνδεσης στην εφαρμογή, καθώς και τον κατάλογο στον οποίο θα πρέπει ν' αναζητά βάσεις δεδομένων. Φυσικά, μπορούμε να μην κάνουμε και καμία ρύθμιση. Σ' αυτή την περίπτωση, όμως, κάθε φορά που θέλουμε να μελετήσουμε τη βάση του Dionaea θα πρέπει να την αντιγράφουμε στο DocumentRoot, ώστε να την εντοπίζει το phpliteadmin. Αυτό γίνεται εύκολα, εκτελώντας αυτό:

```
root@dionaea:/var/www# cp /opt/dionaea/var/dionaea/logsql.sqlite
/var/www/
```

Μετά από τα προηγούμενα, κάνοντας μια βόλτα από τη διεύθυνση `http://IP-του-honeypot/phpliteadmin.php` και δίνοντας τον default κωδικό σύνδεσης (admin) θα μπορούμε να μελετάμε άμεσα όλα όσα έχει καταγράψει το honeypot.

Επίλογος

Ελπίζουμε η δημιουργία της διαδικτυακής παγίδας για malware να σας φάνηκε ενδιαφέρουσα ιδέα. Πρόκειται για ένα πραγματικά ξεχωριστό project, μακριά από τις καθιερωμένες πρακτικές στον τομέα της ασφάλειας. Εξάλλου, μιλάμε για μια ενασχόληση η οποία μπορεί να μας προσφέρει άφθονες νέες γνώσεις! Το έτος 2012, αλλά και τα επόμενα, χαρακτηρίζονται από τους ειδικούς ως "η εποχή των malware". Ίσως είναι που στο επόμενο χρονικό διάστημα αναμένεται να κυκλοφορήσουν πολλά νέα, κακόβουλα προγράμματα. Εμείς σίγουρα δεν μπορούμε να εμποδίσουμε τη δημιουργία τους. Μπορούμε όμως να εφοδιαστούμε με γνώσεις κι εμπειρία, ώστε να τα αντιμετωπίσουμε κατάλληλα!



*Το phpLiteAdmin θυμίζει το phpMyAdmin. Πρόκειται για ένα web interface που παρέχει άμεση κι εύκολη πρόσβαση σε βάσεις SQLite. Με τη βοήθειά του μπορούμε να εξερευνήσουμε άμεσα *και* τη βάση του Dionaea.*

Στο ΔHACKER Μαΐου...

Διαβάστε μεταξύ άλλων πώς μπορούμε να "αποκρυπτογραφούμε" MD5 hashes με το δικό μας πρόγραμμα σε Perl, μάθετε μερικά από τα μυστικά των regular expressions και επιστρατεύστε τα κατάλληλα εργαλεία για επιθέσεις brute-force!

